Robust Functional Supervised Classification for Time Series

Andrés M. Alonso[1], David Casado[2], Sara López-Pintado[3]
and Juan J. Romo[1]

[1] *Universidad Carlos III de Madrid* [2] *Universidad Complutense de Madrid*

[3] *Columbia University*

Abstract

We propose using the integrated periodogram to classify time series. The method assigns a new time series to the group that minimizes the distance between the time series integrated periodogram and the group mean of integrated periodograms. Local computation of these periodograms allows to apply this approach to nonstationary time series. Since the integrated periodograms are curves, we apply functional data depth-based techniques to make the classification robust which is a clear advantage over other competitive procedures. The method provides small error rates with both simulated and two real data examples, improving on existing approaches, and presents good computational behavior.

Key words and phrases: time series, supervised classification, integrated periodogram, functional data depth.

Accepted at Journal of Classification

https://doi.org/10.1007/s00357-014-9163-x

1 Introduction

Classification of time series is an important tool in several fields. Time series can be studied from both time and frequency domains. For short stationary series, a time domain approach based on usual multivariate techniques can be applied. However, the frequency point of view is particularly important for nonstationary series (Huang, Ombao and Stoffer [2004]) and, thus, our proposal follows a frequency domain approach. There exist many papers on supervised classification methods for stationary processes in both domains (see e.g. references in chapter 7 of Taniguchi and Kakizawa [2000]). Several authors have already proposed methods for discriminating between nonstationary models: By using optimal scoring, Hastie et al. (1995) cast the classification problem into the regression framework, where a penalized technique can be applied to the coefficients: [in these cases] "it is natural, efficient and sometimes essential to impose a spatial smoothness constraint on the coefficients, both for improved prediction performance and interpretability". Their proposal is designed for situations where the discriminant variables (predictors) are highly correlated, e.g. when a function is discretized. The following approaches are based on Dahlhaus's local stationarity framework. In Shumway (2003) the Kullback-Leibler discrimination information measure (it is not a real distance) is used, and it is evaluated by using the smoothed time-varying spectral estimator. For clustering, they consider the symmetrized version of this measure. In a first step, Huang et al. (2004) select from SLEX a basis explaining the difference between the classes of time series as well as possible; in a second step, they construct a discriminant criterion that is related to the SLEX spectra of the different classes: a time series is assigned to the class minimizing the Kullback-Leibler divergence between the estimated spectrum and the spectrum of the class. Sakiyama and Taniguchi (2004) use a consistent classification criterion which is an approximation of the Gaussian likelihood ratio. By introducing and influence function, they investigate the behavior of their measure with respect to infinitesimal perturbations of the spectra. In Hirukawa (2004) the approximation of the measure introduced by Sakiyama and Taniguchi (2004) is generalized to nonlinear time-varying spectral measures (including the Kullback-Leibler and Chernoff discrimination information measures). They also propose another approach for non-Gaussian processes. The discrimination of Chandler and Polonik (2006) is based on some features —shape measures or, better, measures of concentration of the variance function— that are measured for each time series. Since it is not distance-based, their approach does not require aligning the series. Both time and frequency domains are connected in Maharaj and Alonso (2007), who combine the techniques of wavelet analysis with those of discriminant analysis. Other related line of research is unsupervised classification of time series, see Liao (2005) for a comprehensive survey.

In this paper, we propose using the integrated periodogram for classifying (locally stationary) time series. The integrated periodogram has the following properties that improve the classification procedure: i) it is a nondecreasing, smooth curve; ii) it presents good asymptotic properties: while the periodogram is an asymptotically unbiased but inconsistent estimator of the spectral density,

the integrated periodogram is a consistent estimator of the spectral distribution (see, chapter 6 of Priestly, 1981); iii) although for stationary processes the integrated spectrum is usually estimated through the spectrum, from a theoretical point of view, the spectral distribution always exists whereas the spectral density only exists under absolutely continuous distributions.

Since the integrated periodogram is a function, we shall use specific techniques for functional data. There is a vast body of existing literature on the statistical analysis of functional data and, particularly, on their classification. For example, a penalized discriminant analysis is proposed in Hastie, Buja and Tibshirani (1995); it is adequate for situations with many highly correlated predictors, as those obtained by discretizing a function. Nonparametric tools to classify a set of curves have been introduced in Ferraty and Vieu (2003), where the authors calculate the posterior probability of belonging to a given class of functions by using a consistent kernel estimator. A new method for extending classical linear discriminant analysis to functional data has been analyzed in James and Hastie (2001): this technique is particularly useful when only fragments of the curves are observed. The problem of unsupervised classification or clustering of curves is addressed in James and Sugar (2003), who elaborate a flexible model-based approach for clustering functional data; it is effective when the observations are sparse, irregularly spaced or occur at different time points for each subject. In Abraham, Cornillon, Matzner-Løber and Molinari (2003) unsupervised clustering of functions is considered; they fit the data by B-splines and the partition is done over the estimated model coefficients using a k-means algorithm. In a related problem, Hall, Poskitt and Presnell (2001) explore a functional data-analytic approach to perform signal discrimination. Nevertheless, many of these procedures are highly sensitive to outliers. A natural and simple way to classify functions is to minimize the distance between the new curve and a reference function of the group. The technique presented in this paper follows this approach. We first consider the mean of the integrated periodograms as the group representative element and then, as a second approach, we use the idea of "deepest" curves instead of the mean to make the method robust.

The notion of statistical depth has already been extended to functional data (see, e.g., López-Pintado and Romo, 2009). In López-Pintado and Romo (2006) the concept of depth is used to classify curves. A statistical depth expresses the "centrality" or "outlyingness" of an observation within a set of data and provides a criterion to order observations from center-outward. Since robustness is an interesting feature of statistical methods based on depth, we have applied the ideas in López-Pintado and Romo (2006) to add robustness to our time series classification procedure. Their method considers ordering the curves within a sample based on a notion of depth for functions and obtaining the α-trimmed mean as a reference curve of each group.

The paper is organized as follows. In section 2 we include some definitions and describe the classification algorithm based on the integrated periodogram. Section 3 explains how depth can be used to make the method robust. Next two sections, 4 and 5, show the behavior of the procedure with simulated and real data, respectively. A brief summary of conclusions is given in section 6.

2 Classifying Time Series

We propose transforming the initial time series into functional data by considering the integrated periodogram of each time series. This allows us to use functional data classification techniques. Let $\{X_t\}$ be a stationary process with autocovariance function $\sigma_h = cov(X_t, X_{t-h})$, such that $\sum_{h=-\infty}^{+\infty} |\sigma_h| < +\infty$, and autocorrelation function $\rho_h = \sigma_h/\sigma_0$. The spectral density is $f(\omega) = \sum_{h=-\infty}^{+\infty} \rho_h \exp(-2\pi i h\omega)$, and it holds that $\rho_h = \int_{-1/2}^{+1/2} \exp(2\pi i h\omega) dF(\omega)$, where F is the spectral distribution function.

The *periodogram* is the corresponding sample version of the spectral density and expresses the contribution of the frequencies to the variance of the series. Let $X = (x_1, \ldots, x_T)$ be a time series. The periodogram is

$$I_T(\omega_k) = \sum_{h=-(T-1)}^{(T-1)} \hat{\rho}_h \exp(-2\pi i h\omega_k), \tag{1}$$

where $\hat{\rho}_h$ denotes the sample autocorrelation at lag h and ω_k takes values in $\{k/T \mid k = 0, \ldots, [T/2]\}$, the discrete *Fourier frequencies* set. Its cumulative version is the *integrated or cumulative periodogram* $F_T(\omega_k) = \sum_{i=1}^{k} I_T(\omega_i)$. The normalized version is

$$F_T(\omega_k) = \sum_{i=1}^{k} I_T(\omega_i) / \sum_{i=1}^{m} I_T(\omega_i), \tag{2}$$

where m is the number of Fourier frequencies. Notice that the denominator in (2) is proportional to the variance of the time series since $2\sum_{i=1}^{m} I_T(\omega_i) = \sum_{t=1}^{T}(x_t - \bar{x})^2$. Therefore, the nonnormalized version of the cumulative periodogram consider not only the shape of the integrated spectrum but also the scale. The normalized version of the cumulative periodogram emphasizes the shape of the curves instead of the scale. For instance, if two time series have spectral densities such that $f_X(\omega) = cf_Y(\omega)$ for some $c > 1$, then these series will have different integrated periodograms but equal normalized integrated periodograms. See Diggle and Fisher (1991) for details on the comparison of cumulative periodograms. As a simple criterion we propose using the normalized version of the cumulative periodogram when the graphs of the functions of the different groups tend to intersect inside their domain of definition. If this is not the case, we recommend using the nonnormalized version. Notice also that the integrated periodogram is a consistent estimator of the integrated spectrum (see, v.g., chapter 6 of Priestley (1981)).

Definitions (1) and (2) correspond to some particular values of ω, but they can be extended to any value in the interval $(-1/2, +1/2)$. Since the periodogram is defined only for stationary series, to classify nonstationary time series we will consider them as locally stationary; this allows us to split the series into blocks, compute the integrated periodogram of each block and merge these periodograms in a final curve: the idea is to approximate the locally stationary processes by piecewise stationary processes. Figure 2(b) provides a blockwise spectral distribution estimation of the locally stationary process spectrum. There are two opposite effects when we increase the numbers of blocks: first, we get closer to the locally stationarity assumption; second, the integrated

periodogram becomes a worse estimator of the integrated spectrum. Notice that this blockwise approach is compatible with the locally stationary time series model of Dahlhaus (1997) where an increasing T implies that more and more data of local structures are available, i.e., we can consider the number of blocks as an increasing function of T. In the appendix, we present the locally stationary model of Dahlhaus (1997) and we propose an integrated spectrum based on this model. This integrated spectrum can be considered as a population version of our blockwise integrated spectrum.

A simple criterion to classify functions is to assign a new observation to the group to which, on the basis of some distance, the function is nearest. In our context, we propose to classify a new series in the group minimizing the distance between the integrated periodogram of the series and a reference curve from the group. We first consider the group mean as a reference curve. If $\Psi_{gi}(\omega)$, $i = 1, \ldots, N$, are functions of group g, the mean is

$$\bar{\Psi}_g = \frac{1}{N} \sum_{i=1}^{N} \Psi_{gi}(\omega). \tag{3}$$

In our case, $\Psi_{gi}(\omega)$ is the concatenated integrated periodogram of the ith series in group g. To measure proximity, we have chosen the L_1 distance,

$$\begin{aligned} d(\Psi_1, \Psi_2) &= \int_{-1/2}^{+1/2} |\Psi_1(\omega) - \Psi_2(\omega)| d\omega \\ &= \sum_{j=1}^{k} \int_{-1/2}^{+1/2} |F_1^{(j)}(\omega) - F_2^{(j)}(\omega)| d\omega, \end{aligned} \tag{4}$$

where k is the number of blocks in which the time series is divided and $F^{(j)}$ is the integrated periodogram of the jth block. The integrated periodograms belong to the $L_1[-1/2, +1/2]$ space. Some other distances could have also been considered. For example, the L_2 distance would highlight large differences between functions.

Based on these definitions we introduce the following classification algorithm:

Algorithm 1

Let $\{X_1, \ldots, X_M\}$ be a sample containing M series from population P_X and let $\{Y_1, \ldots, Y_N\}$ be a sample containing N series from P_Y. The classification method includes the following steps:

1. Split each series into k stationary blocks, calculate the integrated periodogram in each block, and merge the integrated periodograms: $\{\Psi_{X_1}, \ldots, \Psi_{X_M}\}$ and $\{\Psi_{Y_1}, \ldots, \Psi_{Y_N}\}$, where $\Psi_{X_i} = (F_{X_i}^{(1)} \ldots F_{X_i}^{(k)})$, $\Psi_{Y_i} = (F_{Y_i}^{(1)} \ldots F_{Y_i}^{(k)})$, and $F_{X_i}^{(j)}$ is the integrated periodogram of the jth block of the ith series of population X; and analogously for Y. Figures 2(b) and 6 illustrate the obtained Ψ_{X_i}.
2. Calculate the corresponding group means, $\bar{\Psi}_X$ and $\bar{\Psi}_Y$.

3. Let $\Psi_Z = (F_Z^{(1)} \dots F_Z^{(k)})$ be the integrated periodogram of a new series Z. Z is classified in the group P_X if $d(\Psi_Z, \bar{\Psi}_X) < d(\Psi_Z, \bar{\Psi}_Y)$; and in the group P_Y, otherwise.

Remark 1 *Set $k = 1$ to apply the algorithm to stationary series. For nonstationary series, we have used a dyadic splitting of the series into blocks in the simulation and real data computations, $k = 2^p$, $p = 0, 1, \dots$. The implementation with blocks of different lengths, as suggested by visual inspection of data, is also possible. To select the number of blocks, our code implements an optional nested/secondary cross-validation loop to select in each run the value of k that minimizes the global error (we register these values during the runs to form weights that can be thought as relative frequencies). When the previous loop is not called, the minimum global error of each run is registered and the user is given an estimation of the error that would have arised if the number of blocks had been optimized. For this loop to be applicable to small real data sets, the training data of the primary cross-validation loop are used for both optimizing the number of blocks and estimating the final misclassification error rates.*

Remark 2 *Although we are considering $G = 2$, the classification method is obviously extended to the general case in which there are G different groups or populations P_g, $g = 1, \dots, G$.*

Remark 3 *The same methodology could be implemented by using different classification criteria between curves, reference functions for each group (as we do in the following section) or distances between curves.*

Remark 4 *Notice that in this paper, we only consider nonstationarities in the autocovariance structure but we assume that the series are mean stationary. In the case of nonstationaries in the mean (trends, level shifts, piecewise trends, etc.), we should divide the analysis in two cases: (*i*) The nonstationaries in the mean are different in the two populations so they will be useful to improve the classification procedure. In this case, a possibility is to follow the admissible linear procedure described in section 7.2.3 of Taniguchi and Kakizawa (2000) but this is out of the scope of this paper. (*ii*) The nonstationaries in the mean are equal in the two populations so they will not be useful to improve the classification procedure. In this case, we should remove the nonstationaries in the mean by, for instance, the Hodrick-Prescott filter (see Hodrick and Prescott, 1997) or the detrending procedure based on Loess (see Cleveland et al, 1990).*

3 Robust Time Series Classification

Our classification method depends on the reference curve used to measure the distance to the group. The mean of a set of functions is not robust to the presence of outliers. Thus, robustness can be added to this technique by using a robust reference curve. Instead of considering the mean of the integrated periodograms in the group, we shall consider the α-trimmed mean, where

only the deepest elements are averaged. This trimming adds robustness by making the reference function more resistant to outliers.

The statistical depth expresses the "centrality" of each element inside the group. Different definitions of depth are already available. In this section we first describe the concept of depth extended to functional data by López-Pintado and Romo (2009) and then we propose a robust version of our classification algorithm.

Let $G(\Psi) = \{(t, \Psi(t)) \mid t \in [a, b]\}$ denote the graph in $\mathbb{R}^2$ of a function $\Psi \in C[a, b]$, the set of real continuous functions on the interval $[a, b]$. Let $\Psi_i(t)$, $i = 1, \ldots, N$, be functions in $C[a, b]$. The functions $\Psi_{i_j}(t)$, $j = 1, \ldots, k$, determine a band in $\mathbb{R}^2$,

$$B(\Psi_{i_1}, \ldots, \Psi_{i_k}) = \{(t, y) \in [a, b] \times \mathbb{R} \mid \min_{r=1,\ldots,k} \Psi_{i_r}(t) \leq y \leq \max_{r=1,\ldots,k} \Psi_{i_r}(t)\} \tag{5}$$

Given a function Ψ,

$$BD_N^{(j)}(\Psi) = \binom{N}{j}^{-1} \cdot \sum_{1 \leq i_1 < i_2 < \ldots < i_j \leq N} I\{G(\Psi) \subset B(\Psi_{i_1}, \ldots, \Psi_{i_j})\} \tag{6}$$

$j \geq 2$, expresses the proportion of bands determined by different curves $\Psi_{i_1}, \ldots, \Psi_{i_j}$ that contain the graph of Ψ (the indicator function takes the value $I\{A\} = 1$ if A occurs, and $I\{A\} = 0$, otherwise). For functions $\Psi_i(t)$, $i = 1, \ldots, N$, the *band depth* of any of these curves Ψ is

$$BD_{N,J}(\Psi) = \sum_{j=2}^{J} BD_N^{(j)}(\Psi), \tag{7}$$

$2 \leq J \leq N$. If $\tilde{\Psi}$ is the stochastic process generating the observations $\tilde{\Psi}_i(t)$, $i = 1, \ldots, N$, the population versions of these indexes are $BD^{(j)}(\Psi) = P\{G(\Psi) \subset B(\tilde{\Psi}_{i_1}, \ldots, \tilde{\Psi}_{i_j})\}$, $j \geq 2$, and $BD_J(\Psi) = \sum_{j=2}^{J} BD^{(j} = \sum_{j=2}^{J} P\{G(\Psi) \subset B(\tilde{\Psi}_{i_1}, \ldots, \tilde{\Psi}_{i_j})\}$, $J \geq 2$, respectively. In order to illustrate the calculation of the *band depth*, consider the following example: Assume that we have two time series generated by AR(1) models:

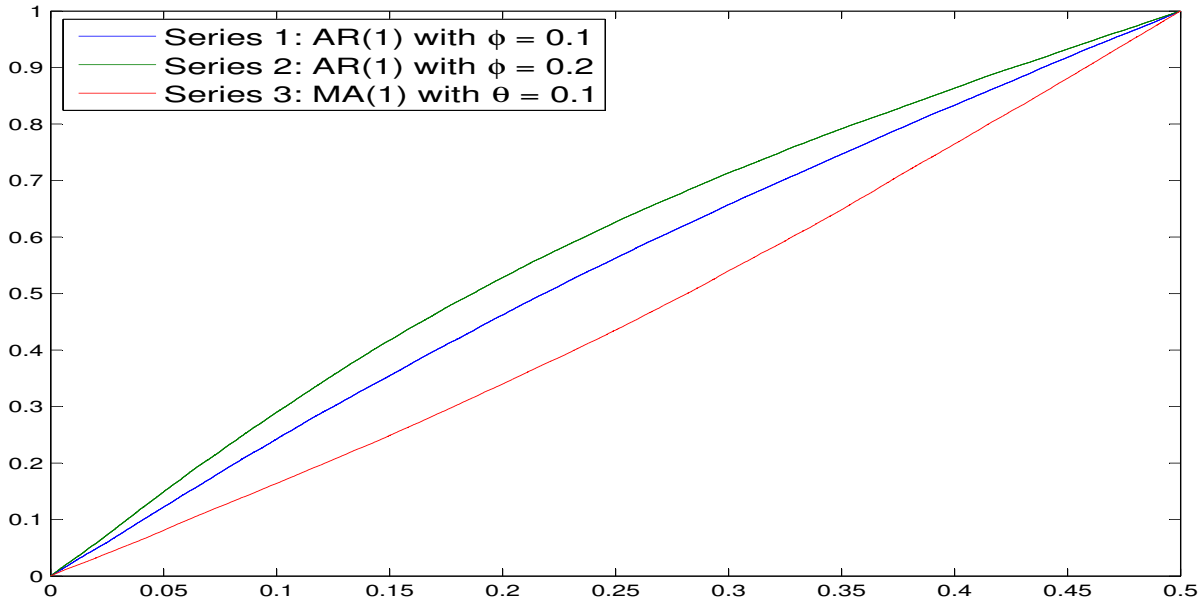
$$X_t^{(i)} = \phi X_{t-1}^{(i)} + \epsilon_t^{(i)},$$

with $\phi^{(1)} = 0.1$, $\phi^{(2)} = 0.2$ and an additional time series generated by a MA(1) model:

$$X_t^{(3)} = \theta \epsilon_{t-1}^{(3)} + \epsilon_t^{(3)},$$

where $\theta = 0.1$ and the $\epsilon_t^{(i)}$ are i.i.d. N(0,1). Figure 1, shows the three (theoretical) integrated periodograms. In order to calculate the depth of each function (integrated periodogram), we determine the $\binom{3}{2} = 3$ bands defined by these three functions, i.e. the bands defined by (1,2), (1,3) and (2,3). Notice that integrated periodogram of the first series is included in the three bands and integrated periodograms two and three are include in only two bands, therefore their band depths are 1, 2/3 and 2/3, respectively. For instance, as the graph shows, the integrated periodogram of the first series is the deepest element.

Figure 1: Example of three integrated periodograms



The *modified band depth* is a more flexible notion of depth defined also in López-Pintado and Romo (2009). The indicator function in (6) is replaced by the length of the set where the function is inside the corresponding band. For any function Ψ of $\Psi_i(t)$, $i = 1, \ldots, N$, and $2 \leq\ j \leq N$, let

$$A_j(\Psi) \equiv A(\Psi; \Psi_{i_1}, \ldots, \Psi_{i_j}) \equiv \{t \in [a,b] \mid \min_{r=i_1,\ldots,i_j} \Psi_r(t) \leq \Psi(t) \leq \max_{r=i_1,\ldots,i_j} \Psi_r(t)\} \quad (8)$$

be the set of points in the interval $[a,b]$ where the function Ψ is inside the band. If λ is the Lebesgue measure on the interval $[a,b]$, $\lambda(A_j(\Psi))$ is the "proportion of time" that Ψ is inside the band. Thus,

$$MBD_N^{(j)}(\Psi) = \binom{N}{j}^{-1} (\lambda[a,b])^{-1} \cdot \sum_{1 \leq i_1 < i_2 < \ldots < i_j \leq N} \lambda(A(\Psi; \Psi_{i_1}, \ldots, \Psi_{i_j})) \quad (9)$$

with $2 \leq j \leq N$, is the generalized version of $BD_N^{(j)}$. If Ψ is always inside the band, the measure $\lambda(A_j(\Psi))$ is 1 and this definition generalizes the definition of depth given in (7). Finally, the modified band depth of any of the curves Ψ in $\Psi_i(t)$, $i = 1, \ldots, N$, is

$$MBD_{N,J}(\Psi) = \sum_{j=2}^{J} MBD_N^{(j)}(\Psi), \quad (10)$$

with $2 \leq J \leq N$. If $\tilde{\Psi}_i(t)$, $i = 1, \ldots, N$, are independent copies of the stochastic process $\tilde{\Psi}$, the population version of these indexes are $MBD^{(j)}(\Psi) = \mathbb{E}\lambda(A(\Psi; \tilde{\Psi}_{i_1}, \ldots, \tilde{\Psi}_{i_j}))$ and $MBD_J(\Psi) = \sum_{j=2}^{J} MBD^{(j)}(\Psi) = \sum_{j=2}^{J} \mathbb{E}\lambda(A(\Psi; \tilde{\Psi}_{i_1}, \ldots, \tilde{\Psi}_{i_j}))$, respectively, for $2 \leq J \leq N$.

Given a sample of functions, $(\Psi_{g_1}, \Psi_{g_2}, \ldots, \Psi_{g_N})$, we can order the sample by calculating the sample modified band depth, $MBD_{N,J}(\Psi_{g_i})$, of each function Ψ_{g_i} for $i = 1, 2, \ldots, N$. The ordered sample is denoted by $(\Psi_{g_{(1)}}, \Psi_{g_{(2)}}, \ldots, \Psi_{g_{(N)}})$, where $\Psi_{g_{(1)}}$ is the deepest function, $\Psi_{g_{(2)}}$ is the second deepest function and so on.

To robustify the algorithm 1, we propose to consider the α-trimmed mean of the group elements as the reference function. If $\Psi_{g_{(i)}}(t)$, $i = 1, \ldots, N$, are functions of the class g ordered by decreasing depth, the α-trimmed mean is

$$\bar{\Psi}_g^{\alpha} = \frac{1}{N - [N\alpha]} \sum_{i=1}^{N-[N\alpha]} \Psi_{g_{(i)}}(t), \tag{11}$$

where $[\cdot]$ is the integer part function. Notice that the median (in the sense of "the deepest") function is also included in the previous expression. We will use $\alpha = 0.2$ in our simulation and real data analysis; it means that for each group the 20% least deep data are not considered when we compute the average.

In step 2 of the new algorithm, the α-trimmed mean replaces the mean as the reference curve for each class. This will make the classification more robust.

Algorithm 2

Let $\{X_1, \ldots, X_M\}$ be a sample containing time series from the population P_X, and let $\{Y_1, \ldots, Y_N\}$ be a sample from P_Y. The classification method includes the following steps:

1. Split each series into k stationary blocks, calculate the integrated periodogram in each block and merge the integrated periodograms: $\{\Psi_{X_1}, \ldots, \Psi_{X_M}\}$ and $\{\Psi_{Y_1}, \ldots, \Psi_{Y_N}\}$, where $\Psi_{X_i} = (F_{X_i}^{(1)} \ldots F_{X_i}^{(k)})$, $\Psi_{Y_i} = (F_{Y_i}^{(1)} \ldots F_{Y_i}^{(k)})$, and $F_{X_i}^{(j)}$ is the integrated periodogram of the jth block of the ith series of the population X; and analogously for Y.
2. Obtain the corresponding group α-trimmed means, $\bar{\Psi}_X^{\alpha}$ and $\bar{\Psi}_Y^{\alpha}$.
3. Let $\Psi_Z = (F_Z^{(1)} \ldots F_Z^{(k)})$ be the integrated periodogram of a new series Z. Z is classified in the group P_X if $d(\Psi_Z, \bar{\Psi}_X^{\alpha}) < d(\Psi_Z, \bar{\Psi}_Y^{\alpha})$, and in the group P_Y, otherwise.

Remark 5 *We have used sample modified band depth with $J = 2$, because this depth is very stable in J, providing similar center-outward ordering in a collection of functions for different values of J (López-Pintado and Romo [2006,2009]).*

Remark 6 *The same algorithm could be implemented using a different functional depth.*

Remark 7 *Computing the depth of functional data is the most time-consuming task in our proposed robust classification algorithm. We implement a preprocessing step to help scale our algorithm to large real data sets as follows. The deepest elements are identified at the beginning so as to maintain only them in the training samples during the runs (although all data are classified). On the one hand, the depth is calculated only once; on the other hand, due to the use of fewer*

but better elements in the training samples, the computational time may be reduced in some cases for which the time spent in calculating the depth is compensated. With this preprocessing step the sizes of the training samples are slightly reduced in most runs, although this has little effect when sample sizes are large. This technique can be applied outside the framework of this work.

MATLAB code is available at `http://www.Casado-D.org/`. Methods *DbC* and *DbC*-α, as well as other characteristics (loop to select the number of blocks, robustifying approach, access to the computational times, etc) are implemented in two scripts, one for the simulation exercises and another for the application to real data. The code is fast and easy to execute and extend. The reader can reproduce, apply or extend our results and plots easily. A help file is also included with the code.

4 Simulation Study

In this section we evaluate our two algorithms and compare them with the method proposed in Huang et al (2004), who use the SLEX (smooth localized complex exponentials) model for a nonstationary random process introduced by Ombao et al (2001). SLEX is a set of Fourier-type bases that are at the same time orthogonal and localized in both time and frequency domains. In a first step, they select from SLEX a basis explaining as good as possible the difference between the classes of time series. After this, they construct a discriminant criterion that is related to the SLEX spectra of the different classes: a time series is assigned to the class minimizing the Kullback-Leibler divergence between the estimated spectrum and the spectrum of the class. For the SLEXbC method we have used an implementation provided by the authors (see `http://hombao.ics.uci.edu/`). To select the parameters, we have performed a small optimization for each simulation and the results were similar to the values recommended to us by the authors.

We have considered the same models as Huang et al (2004). For each comparison of two classes, we run 1000 times the following steps. We generate training and test sets for each model/class. The training sets have the same sizes (sample size and series length) as the ones used by Huang et al (2004), and the test sets contain always 10 series of the length involved in each particular simulation. The methods are tested with the same data sets; this is, in all the models exactly the same simulated time series are used by the three methods, including our algorithms for different values of k.

Simulation 1. We compare an autoregressive process of order one $\{X_t\}$ with Gaussian white noise Y_t:

$$\begin{aligned} X_t^{(i)} &= \phi \cdot X_{t-1}^{(i)} + \epsilon_t^{(i)} \qquad \text{if } \; t = 1, \ldots, T \\ Y_t^{(j)} &= \epsilon_t^{(j)} \qquad\qquad\qquad\quad \text{if } \; t = 1, \ldots, T \end{aligned} \tag{12}$$

with $i = 1, \ldots, M$ and $j = 1, \ldots, N$, where $\epsilon_t^{(i)}$ and $\epsilon_t^{(j)}$ are i.i.d. $N(0,1)$. Each training data set has $M = N = 8$ series of length $T = 1024$. Six comparisons have been run, with the parameter ϕ of the AR(1) model taking the values -0.5, -0.3, -0.1, $+0.1$, $+0.3$ and $+0.5$. Series are stationary in this case.

Simulation 2. We compare two processes composed half by white noise and half by an autoregressive process of order one. The value of the AR(1) parameter is -0.1 in the first class and $+0.1$ in the second class:

$$X_t^{(i)} = \begin{cases} \epsilon_t^{(i)} & \text{if } \ t = 1, \ldots, T/2 \\ X_t^{(i)} = -0.1 \cdot X_{t-1}^{(i)} + \epsilon_t^{(i)} & \text{if } \ t = T/2+1, \ldots, T \end{cases}$$
$$Y_t^{(j)} = \begin{cases} \epsilon_t^{(j)} & \text{if } \ t = 1, \ldots, T/2 \\ Y_t^{(j)} = +0.1 \cdot Y_{t-1}^{(j)} + \epsilon_t^{(j)} & \text{if } \ t = T/2+1, \ldots, T \end{cases} \tag{13}$$

with $i = 1, \ldots, M$ and $j = 1, \ldots, N$. Different combinations of training sample sizes —$M = N = 8$ and 16— and series lengths —$T = 512$, 1024 and 2048— are considered. In this case, the series are made up of stationary parts, but the whole series are not stationary.

Simulation 3. In this case, the stochastic models in both classes are slowly time-varying second order autoregressive processes:

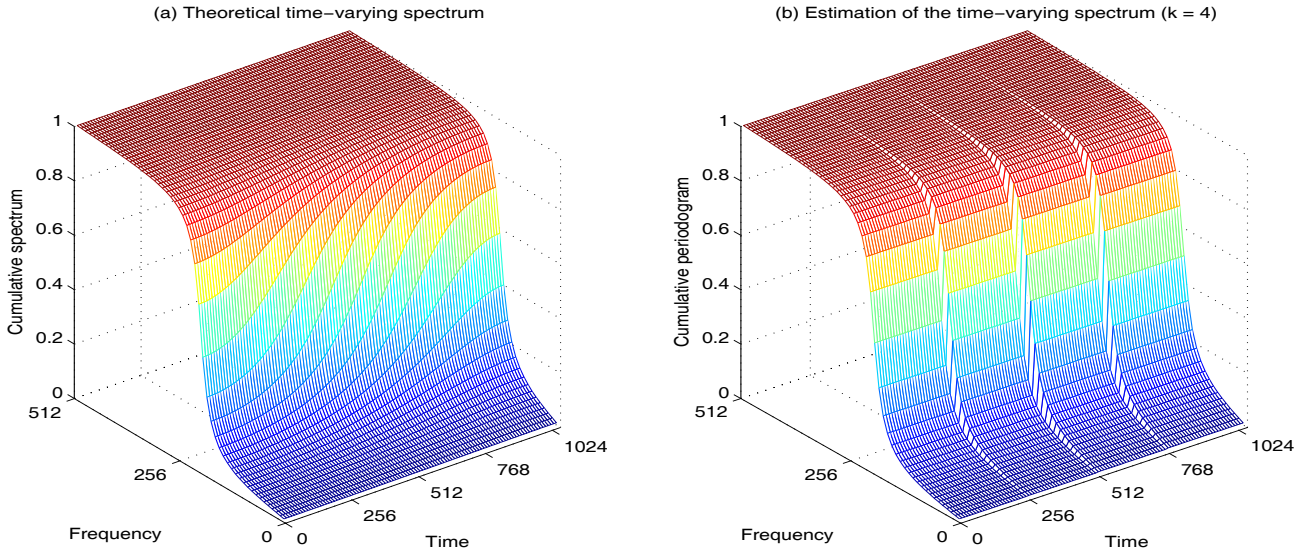
$$X_t^{(i)} = a_{t;0.5} \cdot X_{t-1}^{(i)} - 0.81 \cdot X_{t-2}^{(i)} + \epsilon_t^{(i)} \qquad \text{if } \ t = 1, \ldots, T$$
$$Y_t^{(j)} = a_{t;\tau} \cdot Y_{t-1}^{(j)} - 0.81 \cdot Y_{t-2}^{(j)} + \epsilon_t^{(j)} \qquad \text{if } \ t = 1, \ldots, T \tag{14}$$

with $i = 1, \ldots, M$, $j = 1, \ldots, N$ and $a_{t;\tau} = 0.8 \cdot [1 - \tau \cos(\pi t/1024)]$, where τ is a parameter. Each training data set has $M = N = 10$ series of length $T = 1024$. Three comparisons have been done, the first class having always the parameter $\tau = 0.5$, and the second class having respectively the values $\tau = 0.4$, 0.3 and 0.2. Note that a coefficient of the autoregressive structure is not fixed and it changes with time, making the processes not stationary. See figure 2(a) for an example of the integrated spectrum corresponding to these processes.

We have checked that values between $\tau = -0.9$ and $\tau = +0.9$ do not generate, for any value of t, roots inside the unit circle for the characteristic polynomial of the autoregressive process.

To compare our procedure and the SLEXbC method in terms of robustness, we have performed additional simulations where the training set is contaminated with an outlier time series. In all

Figure 2: Time-varying autoregressive model with $\tau = 0.4$.



cases we contaminate the P_X population by replacing a series by another one following a different model. We consider three levels of contamination: one weak contamination (A) and two strong contaminations (B and C).

Contamination A. For simulation 1, we replace the autoregressive structure for a moving average; that is, we generate a MA(1) model —with the MA parameter equal to the AR parameter— instead of a AR(1) model. For Simulation 2, we change only the autoregressive half of one series in a class (the other half is white noise). For Simulation 3, we contaminate the set of slowly time-varying autoregressives of parameter +0.5 with a series of the same model but with parameter value +0.2.

Contamination B. This type of contamination corresponds to a parameter value of $\phi = -0.9$ in simulations 1 and 2 and $\tau = -0.9$ in simulation 3 instead of the correct value. Therefore, we are always using the correct model except in one case, where we modify the parameter value.

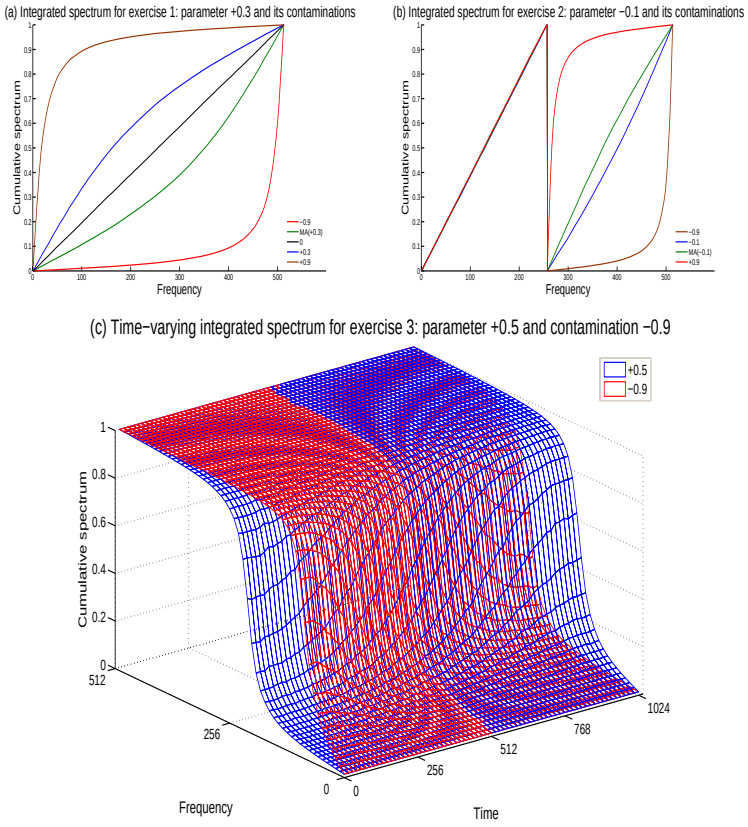
Contamination C. Equal to contamination B, but using a value +0.9 instead of -0.9.

Figures 3(a) and 3(b) present the three contaminations for the first two cases with specific parameter values. Figure 3(c) shows the contamination B for the third case.

The error rates estimates for the first simulation are presented in table 1; for the second simulation, in tables 2, 3, 4 and 5; and for the third simulation in tables 6, 7, 8 and 9. Each cell includes the mean and the standard error (in parenthesis) of the 1000 runs.

Tables 10, 11 and 12 provide the estimates of the computation times. In these tables, each cell includes the mean of the 1000 runs, in seconds. This time is measured for each method from the

Figure 3: Examples of contamination for the three simulation experiments.



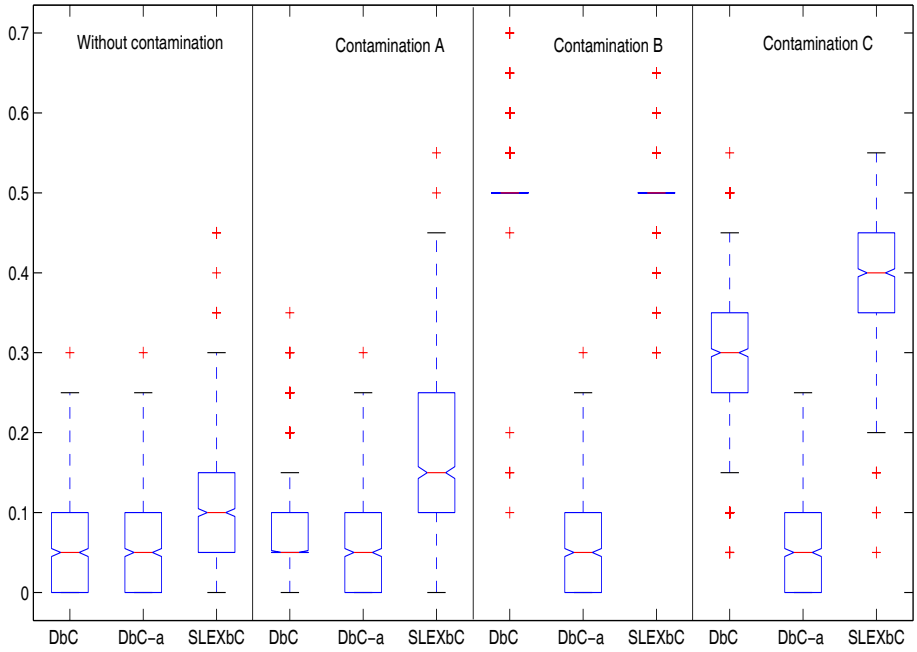
input instant to the moment when the method outputs the error rate. It means that training and test series generation time is not included in the computation; but for our method the computation does include the construction of functional data from series and the evaluation of depth inside groups.

For all tables we use the following notation: DbC (from *depth-based classification*) for algorithm 1, DbC-α for algorithm 2 (using $\alpha = 0.2$ for calculating the α-trimmed mean and $J = 2$ for calculating the modified band depth) and SLEXbC for the method of Huang et al (2004). When a number follows DbC or DbC-α, it indicates the number k of blocks into which the series are split. Given a length T, SLEXbC considers several levels or number of partitions $(1, 2, 2^2, ..., 2^J)$ and usually selects and combines blocks from different levels (that is, blocks of different length) to calculate the SLEX spectrum. For example, for $T = 1024$, partitions into 1,2,4 and 8 blocks are considered by SLEXbC, and the same values have been considered for our methods. The digits in bold correspond to the minima (when they are different to zero).

Comments on Error Rates

Table 1 shows the estimates of the misclassification rates for the first simulation. When contamination is not present, DbC and DbC-α provide similar error rates, and about half of

Figure 4: Boxplot of the misclassification rates in simulation 1, parameter values +0.1 versus 0.



the ones obtained by SLEXbC. As we could expect, for DbC and SLEXbC error rates increase slightly with contamination A (weak) and notably with contaminations B and C (strong), while changes are negligible for DbC-α because the trim keeps the contamination out. DbC error is about half of SLEXbC error for contamination A, but their errors are similar with contaminations B and C. Notice that, in this case, when the autoregressive parameter ϕ is far away from 0 (which corresponds to the second class, Y_t) then the first class, X_t, is separated from the second class. This is why, the three classification methods do not have missclassification for values of $\phi \in \{-0.5, -0.3, 0.3, 0.5\}$. There are some symmetries in table 1 for DbC and SLEXbC: for example, contamination with $\phi = -0.9$ has similar effect on models with ϕ negative/positive than contamination with $\phi = +0.9$ has on models with parameter positive/negative, respectively. To complement the information provided by the tables (mean and standard error), we include some boxplots of the misclassification rates estimates. For simulation 1, we include only the plot of one of the two most difficult comparisons, that is, the comparisons of models $\phi = +0.1$ with Gaussian white noise (see figure 4). The plot shows that SLEXbC tends to have higher median, higher errors above this median, and less errors near zero. On the other side, DbC-α is the only method maintaining the same pattern (with and without contamination) and having a considerable amount of errors close to zero.

Tables 2, 3, 4 and 5 provide the results of the second simulation exercise. As we could expect, errors decrease when any parameter, N or T, increases. Our methods reach the minimum errors when series are divided into two blocks. While our errors are larger than the errors of SLEXbC when we consider the whole series (without splitting them into blocks), errors fall with the first

Table 1: Misclassification rates estimates for simulation 1 with and without contamination.

	$\phi =$ **-0.5**	$\phi =$ **-0.3**	$\phi =$ **-0.1**	$\phi =$ **+0.1**	$\phi =$ **+0.3**	$\phi =$ **+0.5**
Without contamination						
DbC	0.000 (0.0000)	0.000 (0.0000)	**0.063** (0.0017)	**0.060** (0.0017)	0.000 (0.0000)	0.000 (0.0000)
DbC-α	0.000 (0.0000)	0.000 (0.0000)	0.065 (0.0018)	0.062 (0.0017)	0.000 (0.0000)	0.000 (0.0000)
SLEXbC	0.000 (0.0000)	0.000 (0.0000)	0.131 (0.0024)	0.127 (0.0024)	0.000 (0.0000)	0.000 (0.0000)
Contamination A						
DbC	0.000 (0.0000)	0.000 (0.0001)	0.077 (0.0019)	0.074 (0.0019)	0.000 (0.0001)	0.000 (0.0000)
DbC-α	0.000 (0.0000)	**0.000** (0.0000)	**0.064** (0.0017)	**0.062** (0.0017)	**0.000** (0.0000)	0.000 (0.0000)
SLEXbC	0.000 (0.0000)	0.000 (0.0001)	0.175 (0.0028)	0.172 (0.0029)	0.000 (0.0001)	0.000 (0.0000)
Contamination B						
DbC	0.000 (0.0000)	0.000 (0.0001)	0.300 (0.0028)	0.513 (0.0012)	0.001 (0.0002)	0.000 (0.0000)
DbC-α	0.000 (0.0000)	**0.000** (0.0000)	**0.065** (0.0018)	**0.062** (0.0017)	**0.000** (0.0000)	0.000 (0.0000)
SLEXbC	0.000 (0.0000)	0.001 (0.0002)	0.377 (0.0025)	0.491 (0.0011)	0.002 (0.0003)	0.000 (0.0000)
Contamination C						
DbC	0.000 (0.0000)	0.001 (0.0002)	0.512 (0.0013)	0.300 (0.0027)	0.000 (0.0001)	0.000 (0.0000)
DbC-α	0.000 (0.0000)	**0.000** (0.0000)	**0.064** (0.0017)	**0.062** (0.0017)	**0.000** (0.0000)	0.000 (0.0000)
SLEXbC	0.000 (0.0000)	0.002 (0.0004)	0.490 (0.0011)	0.377 (0.0025)	0.001 (0.0002)	0.000 (0.0000)

Table 2: Misclassification rates estimates for simulation 2 without contamination.

	NxT = **8x512**	**16x512**	**8x1024**	**16x1024**	**8x2048**	**16x2048**
DbC 1	0.141 (0.0024)	0.131 (0.0024)	0.062 (0.0017)	0.060 (0.0017)	0.014 (0.0008)	0.014 (0.0008)
2	**0.066** (0.0017)	**0.061** (0.0017)	**0.015** (0.0009)	**0.014** (0.0008)	**0.001** (0.0003)	**0.001** (0.0003)
4	0.078 (0.0019)	0.069 (0.0018)	**0.015** (0.0009)	0.014 (0.0009)	**0.001** (0.0003)	**0.001** (0.0003)
8	0.090 (0.0020)	0.080 (0.0019)	0.020 (0.0010)	0.018 (0.0009)	0.002 (0.0003)	**0.001** (0.0003)
DbC-α 1	0.143 (0.0024)	0.132 (0.0024)	0.063 (0.0017)	0.061 (0.0017)	0.015 (0.0009)	0.014 (0.0008)
2	0.069 (0.0018)	0.064 (0.0017)	0.016 (0.0009)	0.015 (0.0009)	**0.001** (0.0003)	**0.001** (0.0003)
4	0.083 (0.0020)	0.073 (0.0018)	0.017 (0.0010)	0.016 (0.0009)	0.002 (0.0003)	**0.001** (0.0003)
8	0.105 (0.0023)	0.088 (0.0020)	0.024 (0.0011)	0.019 (0.0010)	0.002 (0.0004)	0.002 (0.0003)
SLEXbC	0.114 (0.0023)	0.086 (0.0020)	0.038 (0.0014)	0.025 (0.0011)	0.007 (0.0006)	0.003 (0.0004)

division. As we mentioned before, the length of the blocks decreases with k, and this negatively affects the performance of the periodogram as an estimator. We can observe this effect of splitting in all the tables of simulation 2, and it is also evident that the increase in errors with the increase of k is higher for short series than for longer ones. Nevertheless, we observe that even with $k = 8$ the misclassification rates are smaller than the ones obtained by the SLEXbC procedure or the ones obtained by our procedures with $k = 1$. Recall that, like our procedure, the SLEXbC method splits implicitly the series into blocks. Regarding the contaminations, for DbC and SLEXbC, errors increase slightly with contamination A and greatly for contaminations B and C, while DbC-α maintains its errors and outperforms all the other methods, mainly with strong contaminations and when two blocks are considered. As it could be expected, contaminating a series has major effects when samples sizes are $N_x = N_y = 8$ than when $N_x = N_y = 16$. The DbC and SLEXbC methods are affected more for contamination C than for contamination B, since $\phi = +0.9$ is farther from $\phi = -0.1$ (population P_X) than $\phi = -0.9$ is.

The boxplot error rates from simulation 2 can be seen in figure 5. DbC and DbC-α perform better than SLEXbC. When $k > 1$ the median error rate decreases and it presents a stable behavior. These plots, and the tables, show that DbC-α with $k = 2$ tends to provide the best results, except when there is no contamination and then DbC with $k = 2$ has the best performance. In general, DbC-α with $k = 2$ is the method that presents the largest proportion of errors near zero.

For simulation 3, conclusions similar to the previous ones can be derived from tables 6, 7, 8 and 9. They also show that, in our proposal, penalization for splitting too much is not serious when series are long enough. Generally, the best results with both methods are obtained with $k = 4$ but even with $k = 8$ the misclassification rates are smaller than the ones obtained by the SLEXbC

Table 3: Misclassification rates estimates for simulation 2 with contamination A.

	NxT = 8x512	16x512	8x1024	16x1024	8x2048	16x2048
DbC 1	0.143 (0.0025)	0.132 (0.0024)	0.063 (0.0017)	0.062 (0.0017)	0.018 (0.0010)	0.015 (0.0008)
2	**0.070** (0.0018)	**0.062** (0.0017)	0.018 (0.0010)	**0.014** (0.0008)	0.002 (0.0003)	**0.001** (0.0003)
4	0.083 (0.0020)	0.071 (0.0019)	0.019 (0.0010)	0.015 (0.0009)	0.002 (0.0003)	**0.001** (0.0003)
8	0.102 (0.0022)	0.083 (0.0020)	0.026 (0.0012)	0.019 (0.0010)	0.003 (0.0004)	0.002 (0.0003)
DbC-α 1	0.145 (0.0025)	0.132 (0.0023)	0.063 (0.0017)	0.061 (0.0017)	0.015 (0.0009)	0.014 (0.0008)
2	0.072 (0.0018)	0.064 (0.0017)	**0.015** (0.0009)	0.015 (0.0009)	**0.001** (0.0002)	**0.001** (0.0003)
4	0.086 (0.0021)	0.073 (0.0018)	0.018 (0.0010)	0.016 (0.0009)	0.002 (0.0003)	**0.001** (0.0003)
8	0.114 (0.0024)	0.089 (0.0021)	0.025 (0.0011)	0.019 (0.0010)	0.003 (0.0004)	0.002 (0.0003)
SLEXbC	0.128 (0.0025)	0.092 (0.0021)	0.050 (0.0016)	0.027 (0.0012)	0.012 (0.0008)	0.004 (0.0004)

Table 4: Misclassification rates estimates for simulation 2 with contamination B.

	NxT = 8x512	16x512	8x1024	16x1024	8x2048	16x2048
DbC 1	0.258 (0.0029)	0.168 (0.0026)	0.252 (0.0029)	0.117 (0.0022)	0.250 (0.0029)	0.065 (0.0018)
2	0.135 (0.0024)	0.082 (0.0020)	0.088 (0.0021)	0.030 (0.0012)	0.049 (0.0016)	0.007 (0.0006)
4	0.137 (0.0025)	0.085 (0.0020)	0.089 (0.0021)	0.031 (0.0012)	0.049 (0.0016)	0.007 (0.0006)
8	0.143 (0.0025)	0.092 (0.0021)	0.093 (0.0022)	0.034 (0.0014)	0.050 (0.0016)	0.007 (0.0006)
DbC-α 1	0.145 (0.0024)	0.134 (0.0024)	0.064 (0.0017)	0.061 (0.0017)	0.015 (0.0008)	0.014 (0.0008)
2	**0.070** (0.0018)	**0.065** (0.0017)	**0.017** (0.0010)	**0.015** (0.0009)	0.003 (0.0006)	**0.001** (0.0003)
4	0.081 (0.0020)	0.071 (0.0019)	**0.017** (0.0010)	0.017 (0.0009)	**0.002** (0.0003)	0.002 (0.0003)
8	0.104 (0.0023)	0.087 (0.0020)	0.023 (0.0011)	0.019 (0.0010)	0.002 (0.0004)	0.002 (0.0003)
SLEXbC	0.239 (0.0031)	0.134 (0.0024)	0.228 (0.0030)	0.081 (0.0020)	0.220 (0.0030)	0.037 (0.0013)

Figure 5: Boxplots of the misclassification error rates for simulation 2, training sets with 8 series of length 1024.

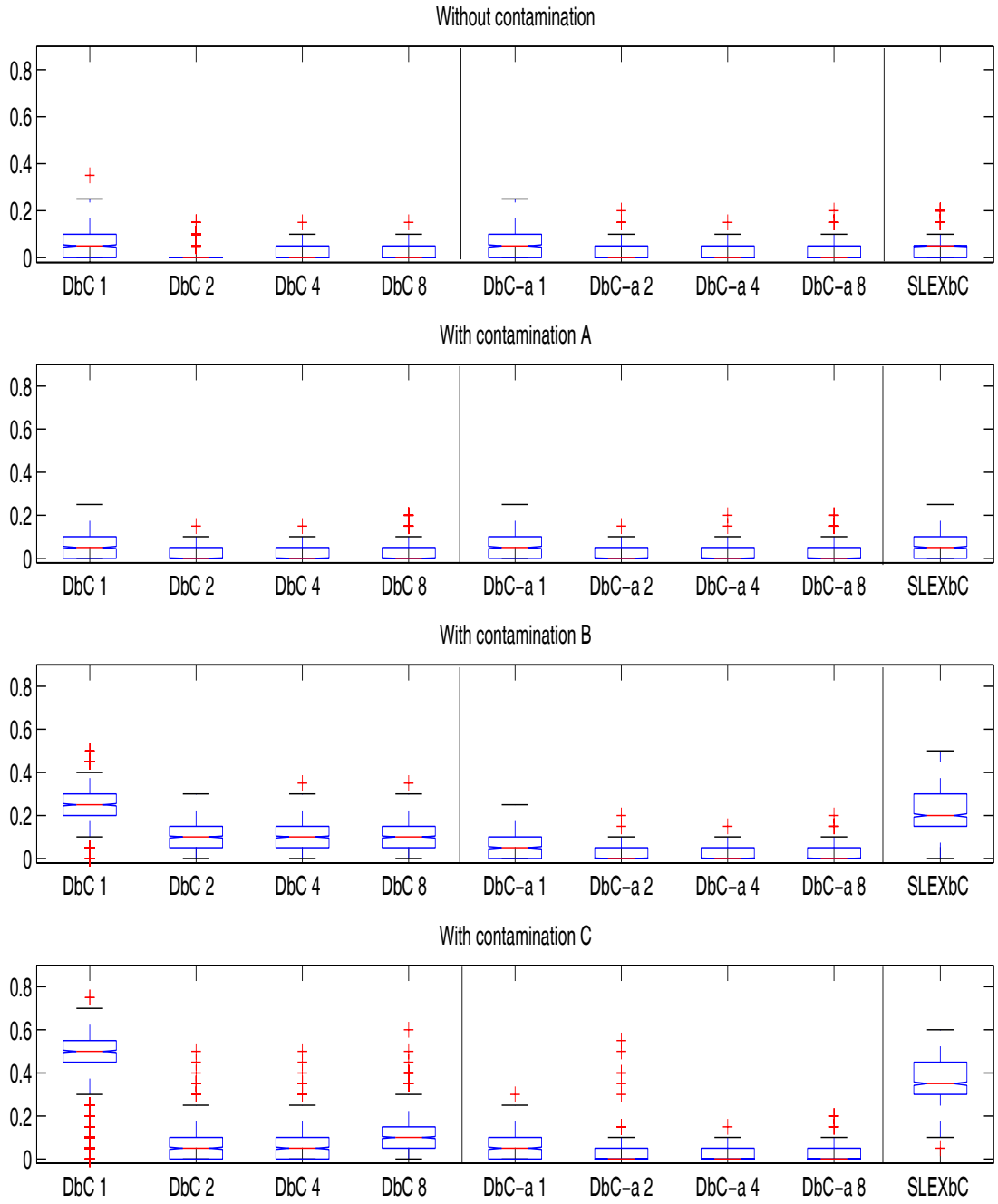


Table 5: Misclassification rates estimates for simulation 2 with contamination C.

	NxT = **8x512**	**16x512**	**8x1024**	**16x1024**	**8x2048**	**16x2048**
DbC 1	0.457 (0.0056)	0.162 (0.0027)	0.437 (0.0055)	0.090 (0.0020)	0.445 (0.0047)	0.038 (0.0013)
2	0.147 (0.0036)	0.078 (0.0019)	0.055 (0.0020)	0.028 (0.0012)	0.015 (0.0010)	0.005 (0.0005)
4	0.187 (0.0037)	0.092 (0.0021)	0.068 (0.0022)	0.030 (0.0012)	0.017 (0.0010)	0.006 (0.0005)
8	0.225 (0.0039)	0.107 (0.0022)	0.101 (0.0027)	0.034 (0.0014)	0.024 (0.0011)	0.006 (0.0006)
DbC-α 1	0.145 (0.0025)	0.133 (0.0024)	0.063 (0.0017)	0.062 (0.0017)	0.015 (0.0009)	0.014 (0.0008)
2	**0.073** (0.0020)	**0.065** (0.0017)	0.018 (0.0013)	**0.015** (0.0009)	0.002 (0.0005)	**0.001** (0.0003)
4	0.083 (0.0020)	0.073 (0.0018)	**0.017** (0.0010)	0.016 (0.0009)	**0.002** (0.0003)	**0.001** (0.0003)
8	0.108 (0.0022)	0.088 (0.0021)	0.024 (0.0011)	0.019 (0.0010)	0.003 (0.0004)	0.002 (0.0003)
SLEXbC	0.376 (0.0036)	0.177 (0.0029)	0.354 (0.0032)	0.098 (0.0023)	0.369 (0.0030)	0.040 (0.0015)

procedure or the ones obtained by our procedures with $k = 1$. Notice that in this case it does not exist a theoretical optimum k. In the contaminated models, the best error rates are obtained with DbC-α for $k = 4$. As we can see, contamination A has a small effect. On the other hand, results are very different for contaminations B and C. Notice that, since τ has positive values in both populations, contaminating with a series of parameter $\tau = -0.9$ (contamination B) is a stronger contamination than using a series with $\tau = +0.9$ (contamination C).

Finally, in the three experiments only a subtle difference can be seen between DbC and DbC-α. When there is no contamination, it is natural that the former provides slightly better error rates, because the latter, due to its trimming, is using only $100(1-\alpha)\%$ of the suitable training data available. Similar results were obtained when the L^2 distance is used instead of L^1. The corresponding tables are available on request to the authors.

Comments on Computation Times

Estimates of the computation times are given in tables 10, 11 and 12. Since chronometer is called after generating series, it can be expected that the computation times do not depend on the parameters of the stochastic processes. This is what we observe for our algorithms, but not for the SLEXbC method. Perhaps, because this method needs to select a basis of the SLEX library for each series, while our method works only with the graphs of the functions (and, at the same time, computing the integrated periodogram does not depend on the parameters).

Some other conclusions that we can point out from the three simulations are the following. It is clear that the computation time for our procedure grows with the number of blocks k. One can also see that the computation of depth is moderately time-consuming with the sample size and

Table 6: Misclassification rates estimates for simulation 3 without contamination.

	$\tau = \mathbf{0.4}$	$\tau = \mathbf{0.3}$	$\tau = \mathbf{0.2}$
DbC 1	0.218 (0.0031)	0.063 (0.0017)	0.019 (0.0010)
2	0.119 (0.0023)	0.006 (0.0006)	**0.000** (0.0000)
4	**0.101** (0.0022)	**0.002** (0.0003)	**0.000** (0.0000)
8	0.123 (0.0024)	0.003 (0.0004)	**0.000** (0.0000)
DbC-α 1	0.226 (0.0032)	0.065 (0.0018)	0.021 (0.0010)
2	0.128 (0.0023)	0.006 (0.0006)	**0.000** (0.0000)
4	0.112 (0.0023)	**0.002** (0.0003)	**0.000** (0.0000)
8	0.139 (0.0026)	0.004 (0.0004)	**0.000** (0.0000)
SLEXbC	0.181 (0.0031)	0.011 (0.0009)	**0.000** (0.0000)

Table 7: Misclassification rates estimates for simulation 3 with contamination A.

	$\tau = \mathbf{0.4}$	$\tau = \mathbf{0.3}$	$\tau = \mathbf{0.2}$
DbC 1	0.232 (0.0032)	0.062 (0.0017)	0.019 (0.0009)
2	0.143 (0.0026)	0.006 (0.0006)	**0.000** (0.0000)
4	0.144 (0.0026)	0.004 (0.0004)	**0.000** (0.0000)
8	0.177 (0.0028)	0.005 (0.0005)	**0.000** (0.0000)
DbC-α 1	0.241 (0.0035)	0.065 (0.0018)	0.020 (0.0010)
2	0.131 (0.0025)	0.007 (0.0006)	**0.000** (0.0000)
4	**0.121** (0.0026)	**0.003** (0.0004)	**0.000** (0.0000)
8	0.150 (0.0029)	0.005 (0.0005)	**0.000** (0.0000)
SLEXbC	0.234 (0.0033)	0.016 (0.0011)	**0.000** (0.0000)

Table 8: Misclassification rates estimates for simulation 3 with contamination B.

	$\tau = \mathbf{0.4}$	$\tau = \mathbf{0.3}$	$\tau = \mathbf{0.2}$
DbC 1	0.254 (0.0029)	0.106 (0.0022)	0.043 (0.0015)
2	0.500 (0.0015)	0.067 (0.0021)	0.001 (0.0002)
4	0.500 (0.0012)	0.062 (0.0020)	0.001 (0.0002)
8	0.499 (0.0013)	0.082 (0.0024)	0.000 (0.0001)
DbC-α 1	0.231 (0.0031)	0.074 (0.0020)	0.026 (0.0012)
2	0.128 (0.0024)	0.007 (0.0006)	**0.000** (0.0000)
4	**0.113** (0.0023)	**0.002** (0.0004)	**0.000** (0.0000)
8	0.141 (0.0026)	0.003 (0.0004)	**0.000** (0.0000)
SLEXbC	0.492 (0.0019)	0.174 (0.0051)	0.015 (0.0009)

Table 9: Misclassification rates estimates for simulation 3 with contamination C.

	$\tau = \mathbf{0.4}$	$\tau = \mathbf{0.3}$	$\tau = \mathbf{0.2}$
DbC 1	0.257 (0.0029)	0.107 (0.0022)	0.044 (0.0015)
2	0.153 (0.0025)	0.017 (0.0009)	0.000 (0.0001)
4	0.128 (0.0024)	0.007 (0.0006)	**0.000** (0.0000)
8	0.132 (0.0024)	0.006 (0.0006)	0.000 (0.0001)
DbC-α 1	0.234 (0.0031)	0.074 (0.0020)	0.025 (0.0012)
2	0.125 (0.0024)	0.007 (0.0006)	0.000 (0.0001)
4	**0.114** (0.0024)	**0.002** (0.0004)	**0.000** (0.0000)
8	0.138 (0.0026)	0.004 (0.0004)	**0.000** (0.0000)
SLEXbC	0.173 (0.0027)	0.015 (0.0009)	0.000 (0.0001)

Table 10: Mean computation times for simulation 1.

	$\phi =$ **-0.5**	$\phi =$ **-0.3**	$\phi =$ **-0.1**	$\phi =$ **+0.1**	$\phi =$ **+0.3**	$\phi =$ **+0.5**
DbC	0.027	0.027	0.027	0.027	0.027	0.027
DbC-α	0.044	0.045	0.045	0.044	0.044	0.044
SLEXbC	0.632	0.678	0.724	0.713	0.670	0.619

series length involved in simulations 1 and 3. This computational complexity comes mainly from the number of comparisons needed when evaluating the depths and not from the length of the series involved in each comparison. Nevertheless, it is possible to do these comparisons only once and speed the computation of depth by implementing conveniently the method in López-Pintado and Romo (2006); this allows using the depths with larger sample sizes. Finally, an interesting effect showed in table 11 is that computation time depends on sample sizes M and N for our approach, but it seems slightly dependent on the series length T, while the SLEXbC method gets slower when any M, N or T increases.

Remark 8 *Notice that computational times depend on the implementation —not just on the method itself—, so we pay closer attention to the qualitative interpretation of the results, as they are less dependent on the programmed code. Additionally, we should mention that results in tables 10, 11 and 12 do not include a k selection procedure for our methods or the smoothing window span for smoothing the SLEX periodogram. Similar qualitative patterns for the computational times have been obtained in simulation exercises —not included here— where the same computer, time series, number of possible values for the parameters and cross-validation loops were considered.*

5 Real data examples

In this section, we illustrate the performance of our proposal in two benchmark data sets: (*i*) Geological data consisting in 16 labeled time series corresponding to eight earthquakes and eight explosion and an unlabeled time series but being an earthquake or an explosion; and (*ii*) Speech recognition data consisting in three sets of 100 labeled time series corresponding to digitized speech frames.

5.1 Geological Data

In this section, we have evaluated our proposal in a benchmark data set containing eight explosions, eight earthquakes and one extra series —known as NZ event—not classified (but being an

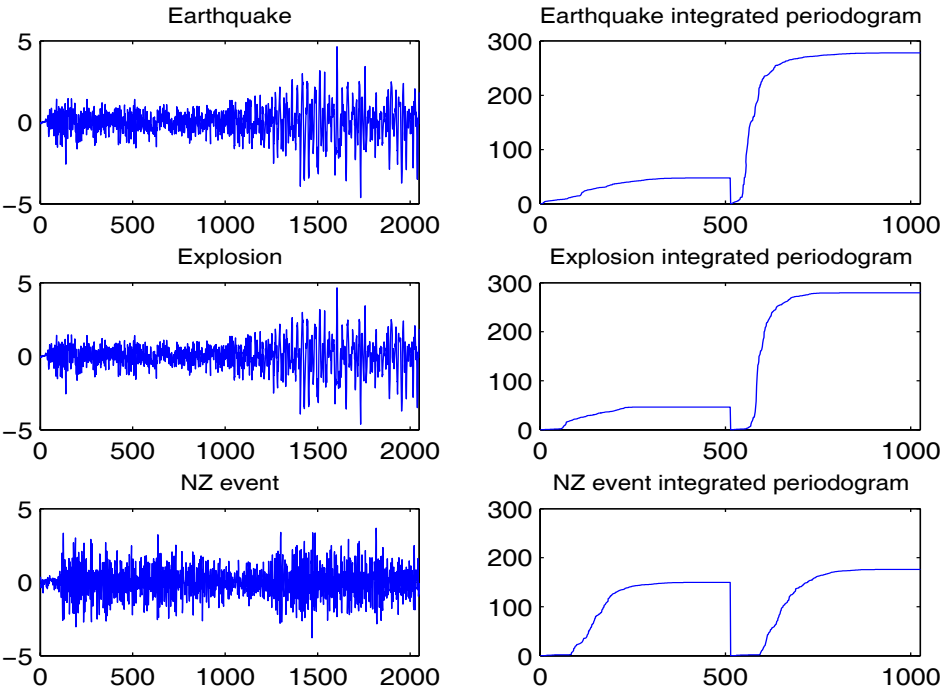
Table 11: Mean computation times for simulation 2.

	NxT = **8x512**	**16x512**	**8x1024**	**16x1024**	**8x2048**	**16x2048**
DbC 1	0.021	0.028	0.027	0.038	0.044	0.067
2	0.036	0.049	0.043	0.060	0.062	0.087
4	0.066	0.092	0.067	0.094	0.081	0.115
8	0.125	0.180	0.126	0.181	0.129	0.186
DbC-α 1	0.031	0.108	0.044	0.200	0.084	0.463
2	0.046	0.137	0.064	0.237	0.103	0.496
4	0.086	0.280	0.087	0.276	0.123	0.505
8	0.170	0.585	0.171	0.595	0.173	0.602
SLEXbC	0.355	0.517	0.736	1.095	1.681	2.506

Table 12: Mean computation times for simulation 3.

	τ = **0.4**	τ = **0.3**	τ = **0.2**
DbC 1	0.031	0.030	0.030
2	0.047	0.047	0.048
4	0.074	0.074	0.074
8	0.140	0.140	0.140
DbC-α 1	0.066	0.062	0.063
2	0.083	0.093	0.094
4	0.120	0.121	0.120
8	0.235	0.234	0.235
SLEXbC	0.733	0.685	0.675

Figure 6: Earthquakes data

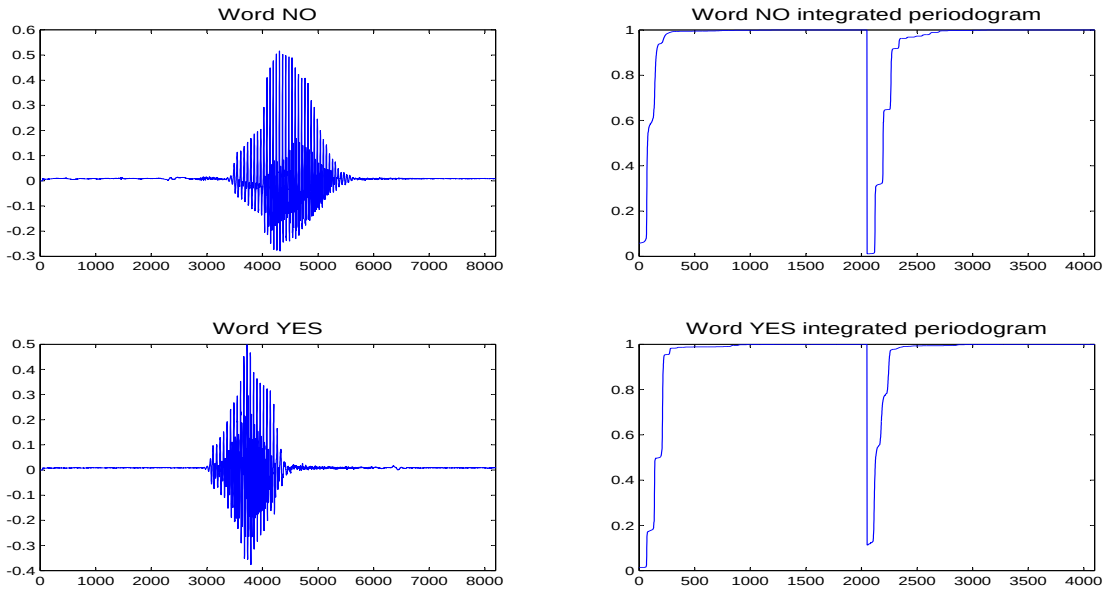


earthquake or an explosion). This data set was constructed by Blandford (1993). Each series contents 2048 points, and its plot clearly shows two different parts — the first half is the part P and the second half is S. This division is an assumption considered by most authors, and it is based on geological reasons. It is also frequent to consider that both parts are stationary. Kakizawa et al. (1998) give a list of these measurements. Shumway and Stoffer (2000) included a detailed study of this data set, and provide access to it at: `http://www.stat.pitt.edu/stoffer/tsa.html`. Figure 6 presents examples of earthquake and explosion, plus the NZ event and their respective integrated periodograms.

Following the simple criterion given in section 2 to choose between normalized and nonnormalized versions of the cumulative periodogram, and after visual observation of these data, for each series we have considered the curve formed by merging the nonnormalized integrated periodograms of parts P and S independently computed; that is, we take $k = 2$ as it is suggested by data (and used by most authors). Let us consider the eight earthquakes as group 1 and the eight explosions as group 2. We have used leave-one-out cross-validation to classify the elements of these two groups: that is, removing a series at a time, using the rest of the data set to train the method and finally classifying the removed series. By doing this, both of our algorithms misclassify the first series of the group 2 (explosions). Regarding the NZ event, if we use the previous groups as training sets, both algorithms agree on assigning it to the explosions group, which agrees with the results obtained by, e. g., Kakizawa et al (1998) or Huang et al (2004).

Now we propose an additional problem. We consider an artificial data set constructed by the eight earthquakes plus the NZ event as group 1, and the eight explosions as group 2. Note that

Figure 7: Words YES/NO data



our method and most of the published papers classify NZ as an explosion. Then we could consider this artificial setting as a case where some atypical observation is presented in group 1. In this situation, the results using our algorithm 1 are that it misclassifies the first and the third elements of group 2 (explosions), not only the first. But again algorithm 2 misclassifies only the first series of group 2. This seems to show the robustness of our second algorithm. Obviously, since we are using leave-one-out cross-validation, both algorithms classify the NZ event in the explosions group, as we mentioned in the previous paragraph.

5.2 Speech recognition data

In this section, we have evaluated our proposal in a benchmark data set containing three subsets of 100 recordings of two short words or phonemes. These three data sets were used by Biau et al. (2003) to illustrate the performance of several classification procedures on functional data, i.e., their procedures consider the time series as functional data. The data are available at `http://www.math.univ-montp2.fr/ biau/bbwdata.tgz`. The first set corresponds to the words YES and NO with 52 and 48 speech frames, respectively; the second set corresponds to the words BOAT and GOAT with 55 and 45 speech frames, respectively; and the third set to the phonemes SH (as in SHE) and AO (as in WATER) with 42 and 58 speech frames, respectively. Each speech frame consists on a time series of length 8192 observations. Figures 7, 8 and 9 present examples of the different words or phonemes and their respective integrated periodograms. As its clear from those figures, the time series are nonstationary so the number of blocks in our procedures should be $k > 1$. For illustrative purpose, we use $k = 2$ in those figures but the "best" k could be selected by a cross-validation procedure.

Biau et al. (2003) report their misclassification rates based on a cross-validation procedure

Figure 8: Words BOAT/GOAT data

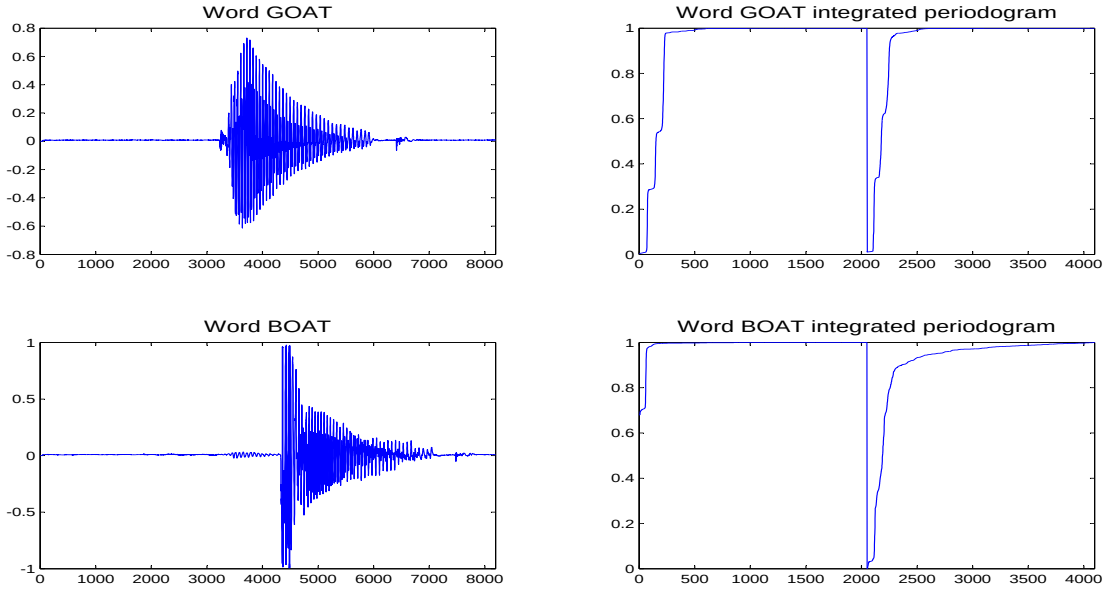


Figure 9: Phonemes SH/AO data

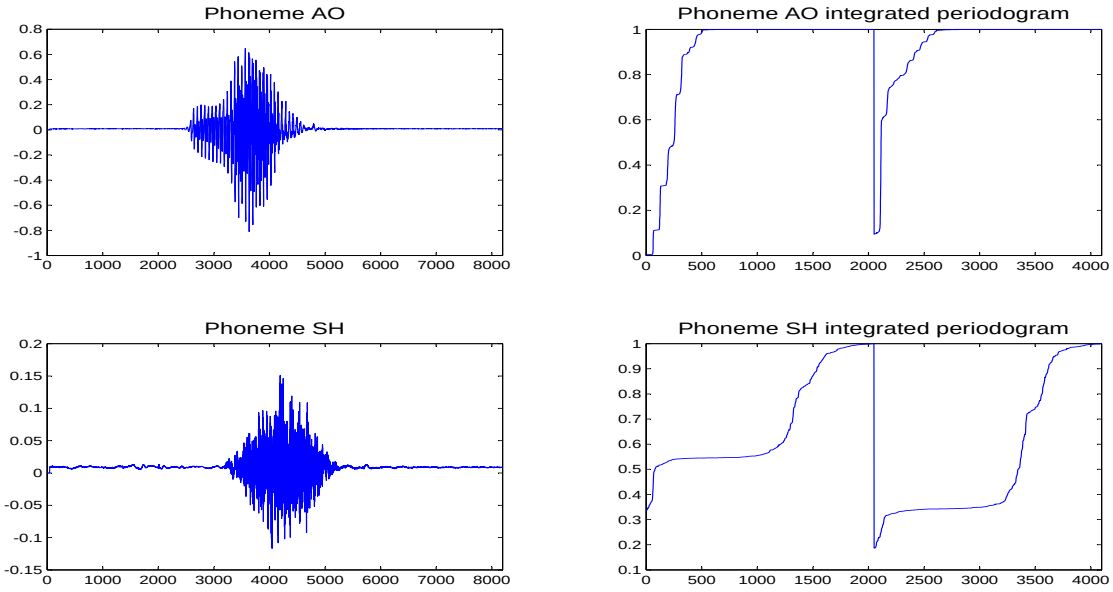


Table 13: Misclassification rates estimates for Speech recognition data.

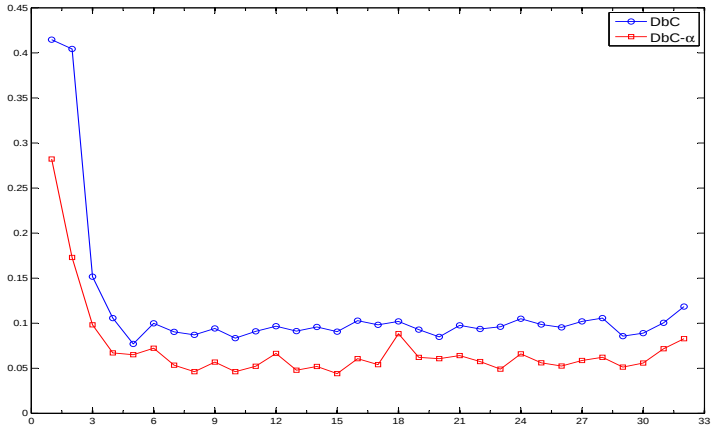
	YES/NO	BOAT/GOAT	SH/AO
DbC 1	0.404 (0.0018)	0.387 (0.0030)	**0.000** (0.0000)
2	0.407 (0.0021)	0.345 (0.0026)	**0.000** (0.0000)
4	0.102 (0.0021)	0.285 (0.0023)	0.003 (0.0002)
8	**0.091** (0.0014)	0.265 (0.0017)	0.003 (0.0002)
16	0.100 (0.0015)	0.253 (0.0028)	0.003 (0.0002)
32	0.117 (0.0016)	**0.250** (0.0028)	0.008 (0.0005)
DbC-α 1	0.281 (0.0038)	0.360 (0.0041)	**0.000** (0.0000)
2	0.170 (0.0032)	0.293 (0.0041)	**0.000** (0.0001)
4	0.066 (0.0012)	0.217 (0.0033)	0.005 (0.0003)
8	**0.049** (0.0009)	0.223 (0.0026)	0.007 (0.0003)
16	0.058 (0.0010)	**0.143** (0.0032)	0.012 (0.0005)
32	0.085 (0.0012)	0.164 (0.0035)	0.023 (0.0006)

with fifty time series as training sample and the remaining fifty time series as testing sample. The results with their nonparametric functional classification procedure and two alternative procedures (nearest neighbor procedure and quadratic discriminant analysis) are 0.10–0.36–0.07, 0.21–0.42–0.35 and 0.16–0.42–0.19 for YES/NO, BOAT/GOAT and SH/AO, respectively. Table 13 shows our classification results using the same cross-validation scheme for different values of k. The misclassification rates estimates reported in table 13 are based on 1000 replications.

Our results are similar or better than the obtained by Biau et al. (2003). The robust algorithm, DbC-α, produces the best results for the YES/NO and BOAT/GOAT sets having misclassification rates around 0.05 (with k=4,8 or 16) and 0.150 (with k=16 or 32), respectively. Both methods, DbC and DbC-α, produce almost perfect classification in the SH/AO set which is a big improvement with respect to the three methods used in Biau et al. (2003). For this third set, the impact of k is not relevant.

Additionally, in figure 10 we show the overall error rate (based on 100 replicas) for the YES/NO dataset using from one to thirty two blocks. The computational time in obtaining figure 10 was around 843.8 seconds, what ratifies the practicability of the number of blocks selection's procedure. Notice that the selection of blocks is performed only once. The best results, using , DbC and DbC-α, are with $k = 5$ and $k = 16$, respectively. Moreover, figure 10 illustrates that, in this dataset, once we select a $k > 4$, the misclassification rates are fairly stable.

Figure 10: Overall error rate estimated by cross-validation in YES/NO dataset.



6 Conclusions

We introduce a new time series classification method based on the series integrated periodogram. Notice that the calculation of the (integrated) periodogram do not involve a bandwidth selection as other spectral (distribution) density estimators. This is a clear advantage with respect to methods that require smooth and consistent spectral density estimators. When the series are nonstationary, they are split into blocks and the integrated periodograms of the blocks are merged to construct a curve; this idea relays on the assumption of local stationarity of the series. Since the integrated periodogram is a function, statistical methods recently developed for functional data can be applied. New series are assigned to the class minimizing the distance between its group mean curve and the new data function. Since the group mean can be affected by the presence of outliers, we propose robustifying the classification by replacing the mean curve by the depth based α-trimmed mean, where for each group only the deepest elements are averaged. We have evaluated our proposal in different scenarios. We have run three simulations containing several models and parameter values, one with stationary series and the other with two types of nonstationarity. After running the simulations without contamination, we have repeated all comparisons three more times using exactly the same series but replacing one by a contaminated series. We consider one kind of weak contamination and two strong contaminations. Our second algorithm exhibits robustness against outlier, meanwhile the performance of the SLEXbC procedure deteriorates considerably. We also illustrate the performance of our procedure in two benchmark datasets. Our proposal provides small error rates, robustness and a good computational behavior, what makes the method suitable for classifying long time series. Finally, this paper suggests that the integrated periodogram contains useful information to classify time series and the concept of depth for functional data can be used to make classification robust, which is a clear advantage over other competitive procedures that are strongly affected by the presence of outliers.

Appendix

In this section we follows the papers of Dahlhaus (1996, 1997) to present a locally stationary time series model that allows us to define a time dependent integrated spectrum. In this nonstationary framework it is not possible to separate the time and the frequency domains. The strategy of Dahlhaus started with a spectral representation:

Definition 1 (Dahlhaus, 1996 and 1997) *A sequence of stochastic processes* $(X_{t,T}\ 1 \leq t \leq T,\ T \geq 1)$ *is called* locally stationary with transfer function A^0 and trend μ *if such a representation exists*

$$X_{t,T} = \mu\left(\frac{t}{T}\right) + \int_{-\pi}^{+\pi} e^{i\lambda t} A^0_{t,T}(\lambda) d\xi(\lambda), \tag{15}$$

where

(i) $\xi(\lambda)$ *is a stochastic process on* $[-\pi, +\pi]$ *with* $\overline{\xi(\lambda)} = \xi(-\lambda)$ *and*

$$cum\{d\xi(\lambda_1), \cdots, d\xi(\lambda_k)\} = \eta(\sum_{j=1}^{k} \lambda_j) g_k(\lambda_1, \cdots, \lambda_{k-1}) d\lambda_1 \cdots d\lambda_k,$$

where $g_1 = 0$, $g_2(\lambda) = 1$, $|g_k(\lambda_1, \cdots, \lambda_{k-1})| \leq const_k$ *for all* k, $cum\{\cdots\}$ *denotes the cumulant of* k*-th order and* $\eta(\lambda) = \sum_{j=-\infty}^{+\infty} \delta(\lambda + 2\pi j)$ *is the period* 2π *extension of the Dirac delta function.*

(ii) There is a constant C *and a* 2π*-periodic function* $A : [0,1] \times \mathbb{R} \to \mathbb{C}$ *with* $A(u, -\lambda) = \overline{A(u,\lambda)}$ *and*

$$sup_{t,\lambda} |A^0_{t,T}(\lambda) - A(t/T, \lambda)| \leq CT^{-1},$$

for all T*;* $A(u,\lambda)$ *and* $\mu(u)$ *are assumed to be continuous in* u*.*

Definition 2 (Dahlhaus, 1996 and 1997) *The* (time-varying) spectral density *of the process (sequence of processes) is defined as:*

$$f(u,\lambda) = A(u,\lambda)\overline{A(u,\lambda)} = |A(u,\lambda)|^2. \tag{16}$$

For these processes, Dahlhaus (1996) also defines the local covariance of lag k at time u, and gives kernel estimates of it, as well as of the spectral density. From the above definition, we can propose a spectral distribution function that could be estimated by our blockwise integrated periodogram:

Definition 3 *The* (time-varying) spectral distribution *of the process (sequence of processes) is defined as:*

$$F(u,\lambda) = \int_{-\pi}^{\lambda} f(u,l) dl. \tag{17}$$

Notice that if the underlying process (sequence of processes) is piecewise stationary, i.e., there exit some $u_1, u_2, \ldots, u_s$ such that $f(u, \lambda)$ is constant as function of u at intervals (u_i, u_{i+1}) then the above definition produces a piecewise constant spectral distribution as function of u. Of course, assuming the asymptotic framework proposed by Dahlhaus, the number of observations in each interval increases as T grows. The integrated periodogram, F_T, calculated using the observations inside a particular interval will provide a consistent estimator of this piecewise spectral distribution. Moreover, if we consider an increasing number of blocks then most of these K intervals will be inside of one the intervals (u_i, u_{i+1}) for $i = 1, 2, \ldots, s-1$ and the length of the intervals that does not satisfy this inclusion property will be asymptotically negligible.

Acknowledgements

Andrés M. Alonso[1,2], David Casado[1], Sara López-Pintado[1], Juan J. Romo[1].

[1] Supported in part by CICYT (Spain) grants SEJ2007-64500, and MICINN (Spain) grant ECO2008-05080.

[2] Supported in part by "Comunidad de Madrid" (Spain) grant S2007/HUM-0413.

References

[1] Abraham, C., Cornillon, P.A., Matzner-Løber, E., and Molinari, N. (2003). Unsupervised Curve Clustering using B-Splines. *Scandinavian Journal of Statistics*. 30 (3), 581–595.

[2] Biau, G., Bunea, F. and Wegkamp, M.H. (2003). Functional Classification in Hilbert Spaces. *IEEE Transactions on Information Theory*. 1 (11), 1–8.

[3] Blandford, R.R. (1993). Discrimination of Earthquakes and Explosions at Regional Distances Using Complexity. *Report AFTAC-TR-93-044 HQ*, Air Force Technical Applications Center, Patrick Air Force Base, FL.

[4] Chandler, G., and W. Polonik (2006). Discrimination of Locally Stationary Time Series Based on the Excess Mass Functional. *Journal of the American Statistical Association*. 101 (473), 240–253.

[5] Cleveland, R.B., Cleveland, W.S., McRae, J.E., and Terpenning, I. (1990). STL: A Seasonal-Trend Decomposition Procedure Based on Loess. *Journal of Official Statistics*. 6, 2–73.

[6] Dahlhaus, R. (1996) Asymptotic Statistical Inference for nonstationary processes with evolutionary spectra. In *Athens Conference on Applied Probability and Time Series Analysis* (P.M. Robinson and M. Rosenblatt, eds.). New York: Springer.

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

[8] Diggle, P.J., and Fisher, N.I. (1991) Nonparametric Comparison of Cumulative Periodograms. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*. 40 (3), 423–434.

[9] Ferraty, F., and P. Vieu (2003). Curves Discrimination: A Nonparametric Functional Approach. *Computational Statistics and Data Analysis*. 44, 161–173.

[10] Fryzlewicz, P., and Ombao, H. (2009). Consistent classification of nonstationary time series using stochastic wavelet representations. *Journal of the American Statistical Association*. 104, 299–312.

[11] Hall, P., D.S. Poskitt and B. Presnell (2001). A Functional Data-Analytic Approach to Signal Discrimination. *Technometrics*. 43 (1), 1–9.

[12] Hastie, T., A. Buja and R.J. Tibshirani (1995). Penalized Discriminant Analysis. *The Annals of Statistics*. 23 (1), 73–102.

[13] Hirukawa, J. (2004). Discriminant Analysis for Multivariate Non-Gaussian Locally Stationary Processes. *Scientiae Mathematicae Japonicae*. 60 (2), 357–380.

[14] Hodrick, R., and Prescott, E.C. (1997). Postwar U.S. Business Cycles: An Empirical Investigation. *Journal of Money, Credit, and Banking*. 29 (1), 1-16.

[15] Huang, H., H. Ombao and D.S. Stoffer (2004). Discrimination and Classification of Nonstationary Time Series Using the SLEX Model. *Journal of the American Statistical Association*. 99 (467), 763–774.

[16] James, G.M., and T. Hastie (2001). Functional Linear Discriminant Analysis for Irregularly Sampled Curves. *Journal of the Royal Statistical Society*. Series B, 63, 533–550.

[17] James, G.M., and C. A. Sugar (2003). Clustering for Sparsely Sampled Functional Data. *Journal of the American Statistical Association*. 98 (462), 397–408.

[18] Kakizawa, Y., R.H. Shumway and M. Taniguchi (1998). Discrimination and Clustering for Multivariate Time Series. *Journal of the American Statistical Association*. 93 (441), 328–340.

[19] Liao, T.W. (2005). Clustering of time series data survey. *Pattern Recognition*. 38, 1857–1874.

[20] López-Pintado, S., and J. Romo. (2006). Depth-Based Classification for Functional Data. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, American Mathematical Society. Vol. 72.

[21] López-Pintado, S., and J. Romo (2009). On the Concept of Depth for Functional Data. *Journal of the American Statistical Association*, 104 (486), 704-717.

[22] Maharaj, E.A., and A.M. Alonso (2007). Discrimination of Locally Stationary Time Series Using Wavelets. *Computational Statistics and Data Analysis*. 52, 879–895.

[23] Ombao, H.C., J.A. Raz, R. von Sachs and B.A. Malow (2001). Automatic Statistical Analysis of Bivariate Nonstationary Time Series. *Journal of the American Statistical Association*. 96 (454), 543–560.

[24] Priestley, M. (1965). Evolutionary Spectra and Nonstationary Processes. *Journal of the Royal Statistical Society*. Series B, 27 (2), 204–237.

[25] Priestley, M. (1981). *Spectral Analysis and Time Series. Volume 1: Univariate series*. London: Academic Press, Inc.

[26] Sakiyama, K., and M. Taniguchi (2004). Discriminant Analysis for Locally Stationary Processes. *Journal of Multivariate Analysis*. 90, 282–300.

[27] Shumway, R.S. (2003). Time-Frequency Clustering and Discriminant Analysis. *Statistics & Probability Letters*. 63, 307–314.

[28] Shumway, R.H., and D.S. Stoffer (2000). *Time Series Analysis and Its Applications*. New York: Springer.

[29] Taniguchi, M., and Y. Kakizawa (2000). *Asymptotic Theory of Statistical Inference for Time Series*. New York: Springer.