

Robust Clustering for Time Series Using Spectral Densities and Functional Data Analysis

Diego Rivera-García¹, Luis Angel García-Escudero², Agustín Mayo-Iscar²,
and Joaquín Ortega¹(✉)

¹ Centro de Investigación en Matemáticas, Guanajuato, Mexico
{driver,jortega}@cimat.mx

² Universidad de Valladolid, Valladolid, Spain
{lagarcia,agustinm}@eio.uva.es

Abstract. In this work a robust clustering algorithm for stationary time series is proposed. The algorithm is based on the use of estimated spectral densities, which are considered as functional data, as the basic characteristic of stationary time series for clustering purposes. A robust algorithm for functional data is then applied to the set of spectral densities. Trimming techniques and restrictions on the scatter within groups reduce the effect of noise in the data and help to prevent the identification of spurious clusters. The procedure is tested in a simulation study, and is also applied to a real data set.

Keywords: Time series clustering · Functional Data Analysis · Clustering · Robustness · Trimming · Functional principal components analysis

1 Introduction

Time series clustering has become a very active research area, with applications in many different fields. However, most methods developed so far do not take into account the possible presence of contamination by outliers or spurious information. In this work, we propose a clustering algorithm for stationary time series that is based on considering the estimated spectral density functions as functional data. This procedure has robust features that mitigate the effect of noise in the data and help to prevent the identification of spurious clusters.

Comprehensive revisions of the area can be found in [1, 8, 19]. [21] present a package in R for time series clustering with a wide range of alternative methods. According to Liao [19], there are three approaches to clustering of time series: methods that depend on the comparison of the raw data, methods based on models fitted to the data and, methods based on features derived from the time series. Our proposal falls within the third approach and the spectral density is the characteristic used to gauge the similarity between time series in the sample.

Spectral characteristics have been previously considered as the main tool for time series clustering. Caiado et al. [6, 7] use the periodogram and normalized periodogram ordinates for clustering time series. Maharaj and D’Urso [20] propose a fuzzy clustering algorithm based on the estimated cepstrum, which is the

spectrum of the logarithm of the spectral density. Alvarez-Esteban et al. [3] and Euán et al. [13] consider the use of the total variation distance on normalized estimates of the spectral density as a similarity measure for clustering. A brief description of the last two algorithms will be given in Sect. 2.

Other works have focused on developing robust clustering algorithms for time series. D’Urso et al. [11] use a fuzzy approach to propose a robust clustering model based on autoregressive models. D’Urso et al. [12] present robust fuzzy clustering schemes for heteroskedastic time series based on parametric models, Bahadori et al. [4] propose a clustering framework for functional data, which can be applied to time series with warped alignments.

Our proposal is based on the use of spectral densities, considered as functional data, and the application of the clustering algorithm recently developed in [24], which will be described in Sect. 3. Several clustering methods for functional data have been proposed in the literature as, for instance, [5, 17, 18] but these methods are not aimed at dealing with outlying curves. Trimming techniques for robust clustering have been applied in [9, 15].

The rest of the paper is organized as follows: Sect. 2 describes the idea behind our proposal for time series clustering. Section 3 gives a brief description of the robust clustering procedure for functional data that supports the time series clustering algorithm. Section 4 presents a simulation study that compares the performance of the algorithm with existing alternatives and Sect. 5 gives an application to a real data set. The paper ends with a discussion of the results.

2 Time Series Clustering

Consider a collection of n stationary time series $X_{1,t}, X_{2,t}, \dots, X_{n,t}$ with $1 \leq t \leq T$. For ease of notation we take all series to have the same length, but this is not a requirement of the procedure. The spectral density of each time series is estimated by one of the many procedures available. Previous clustering methods based on the use of spectral densities relied on similarity measures for discriminating between them. In this work, the spectra are considered as functional data to which the robust clustering procedure developed in [24] is applied. The resulting clusters correspond to time series whose spectral densities have similar shapes, and hence similar oscillatory behavior. The procedure is able to detect outliers among the spectral densities, which correspond to time series having atypical oscillatory characteristics.

Two methods based on estimated spectral densities are presented in [3, 13]. We describe them in more detail since they will be used later for comparison purposes. We refer to them as “TVDClust” and “HSMClust”, respectively. In both cases the total variation distance (TVD) is used to measure similarity between spectral densities. TVD is a frequently-used distance between probability measures that, in the case of probability distributions having a density, measures the complement of the common area below the density curves. Thus, the more alike the densities are, the larger this common area and the smaller the TV distance. To use this distance to compare spectral densities, they need to be normalized so

that the total area below the curve is equal to 1, which is equivalent to normalizing the original time series so that it has unit variance. Thus, it is the oscillatory behavior of the series, and not the magnitude of the oscillations that is taken into account in these clustering algorithms.

For “TDVClust”, a dissimilarity matrix is built up by measuring the TVD distance between all pairs of normalized estimated spectral densities. This matrix is then fed to a hierarchical agglomerative algorithm with the complete or average linkage functions. The result is a dendrogram which can be cut to obtain the desired number of groups. To decide on the number of clusters an external criteria such as the Silhouette or Dunn’s index is used. More details can be found in [3].

The second method, “HSMClust”, is a modification in which every time two clusters are joined together, all the information in them is used to obtain a representative spectrum for the new cluster. There are two ways to do this, either all the spectral densities are averaged, which is the *average* option in the algorithm, or else all the time series in the two groups are concatenated and a new spectral density is estimated, which corresponds to the *single* option. Under the assumption that the series in the same cluster have common oscillatory characteristics, either of this procedures will give a more accurate estimation of the common spectral density for the whole group. This algorithm is known as the Hierarchical Spectral Merger (HSM) algorithm, and its implementation in R is available at <http://ucispacetime.wix.com/spacetime#!project-a/cxl2>.

Every time two clusters are merged, the dissimilarity matrix reduces its size. In “TVDClust”, this matrix remains the same throughout the procedure and the distances between clusters are calculated using linear combinations of the distances of the individual points in each cluster. The linear combination used is determined by the linkage function employed. More details can be found in [3].

3 Robust Clustering for Functional Data

We now give a brief description of the algorithm proposed in [24], where more details can be found. Let X be a random variable taking values in the Hilbert space $L^2([0, T])$ of square integrable functions defined in the interval $[0, T]$, with inner product given by $\langle f, g \rangle = \int f(t)g(t) dt$. If $\mu(t) = E\{X(t)\}$ and $\Gamma(s, t) = \text{cov}\{X(s), X(t)\}$, then it is usual to represent X through its Karhunen-Loève expansion $X(t) = \mu(t) + \sum_{j=1}^{\infty} C_j(X)\psi_j(t)$. In that expansion, the ψ_j are an orthonormal system of functions obtained as eigenfunctions of the covariance operator Γ , i.e. $\langle \Gamma(\cdot, t), \psi_j \rangle = \lambda_j \psi_j(t)$, and the eigenvalues λ_j are taken in decreasing order and assumed to satisfy $\sum_{j=1}^{\infty} \lambda_j < \infty$. The principal component scores $C_j(X) = \langle X - \mu, \psi_j \rangle$ are uncorrelated univariate random variables with zero mean and variance equal to λ_j . Delaigle and Hall [10] show that $\log P(\|X - x\| \leq h)$ can be approximated by $\sum_{j=1}^p \log f_{C_j}(c_j(x))$, for any $x \in L_2([0, T])$ and small h , where f_{C_j} corresponds to the probability density function of $C_j(X)$ and $c_j(x) = \langle x, \psi_j \rangle$. This approximation entails a kind of “small-ball pseudo-density” approach for Functional Data Analysis by taking into account that probability density functions in the finite dimensional case can be seen as the

limit of $P(\|X - x\| \leq h)/h$ when $h \rightarrow 0$. In the particular case of X being a Gaussian process, the $C_j(X)$ are independent normally distributed random variables with mean equal to 0 and variance equal to λ_j .

With these ideas in mind, Jacques and Preda [17] proposed a “model-based” approach for clustering of functional data, where a finite number of independent normally distributed principal component scores are assumed and different variances are also allowed for each cluster. Previously, Bouveyron and Jacques [5] had already considered a different approach, where a certain fraction of the smallest variances are constrained to be equal for each cluster.

In Rivera-García et al. [24], starting from Bouveyron and Jacques [5] and Jaques and Preda [17], a robust functional clustering procedure is proposed where a proportion α of curves are allowed to be trimmed and constraints on the variances are considered. If $\{x_1, \dots, x_n\}$ is a set of curves in $L^2([0, T])$, we consider the maximization of a trimmed mixture-loglikelihood defined as

$$\sum_{i=1}^n \eta(x_i) \log \left(\sum_{g=1}^K \pi_g \left[\prod_{j=1}^{q_g} \frac{1}{\sqrt{2\pi a_{jg}}} \exp \left(\frac{-c_{ijg}^2}{2a_{jg}} \right) \prod_{j=q_g+1}^p \frac{1}{\sqrt{2\pi b_g}} \exp \left(\frac{-c_{ijg}^2}{2b_g} \right) \right] \right) \tag{1}$$

where $c_{ijg} = c_{jg}(x_i)$ is the j -th principal component score of curve x_i in group g , $g = 1, \dots, K$, and, $\eta(\cdot)$ is an indicator function with $\eta(x_i) = 0$ if the x_i curve is trimmed and 1 if it is not and $\pi_g, g = 1, \dots, K$ are mixing weights that add up to 1. A proportion α of curves is trimmed, so that $\sum_{i=1}^n \eta(x_i) = [n(1 - \alpha)]$. The main variance contributions in the g -th cluster are assumed to be a_{1g}, \dots, a_{q_gg} , for the first q_g components, while we assume that each of the remaining $p - q_g$ components contribute with the same b_g variance. Notice that we take an equal number of principal components p in every cluster but the number of main components q_g may vary across clusters. Finally, to prevent the detection of spurious clusters, two constants $d_1 \geq 1$ and $d_2 \geq 1$ were fixed such that the maximization of (1) is done under the constraints:

$$\frac{\max_{g=1, \dots, K; j=1, \dots, q_j} a_{jg}}{\min_{g=1, \dots, K; j=1, \dots, q_j} a_{jg}} \leq d_1 \quad \text{and} \quad \frac{\max_{g=1, \dots, K} b_g}{\min_{g=1, \dots, K} b_g} \leq d_2. \tag{2}$$

A feasible algorithm for performing the constrained maximization, detailed in [24], is a modification of the traditional EM algorithm used in model-based clustering where a “trimming” step (T-step) is added. In the T-step, those curves with smallest contributions to the trimmed likelihood are temporarily not taken into account in each iteration of the algorithm. The trimming step is similar to that applied in the “concentration” steps applied when performing the fast-MCD algorithm [25]. To enforce the required constraints on the variances, optimally truncated variances as done in [14] are adopted if needed. For the estimation of the dimension q_g in each cluster, a Bayesian Information Criterion (BIC) approach was proposed in [24].

4 Simulation Study

To evaluate the performance of the proposed Robust Functional Clustering (RFC) methodology, a simulation study was carried out. We now describe the different scenarios and contamination types. As in [13], the simulations are based on combinations of autoregressive processes of order 2, AR(2), which are defined as $X_t = u_1 X_{t-1} + u_2 X_{t-2} + \epsilon_t$ where ϵ_t is a white noise process. The associated characteristic polynomial is $h(y) = 1 - u_1 y - u_2 y^2$ and its roots, denoted by y_1 and y_2 , are related to the spectrum of the time series. If the roots are complex-valued, they must be conjugate, i.e. $y_1 = \overline{y_2}$ and their polar representation is $|y_1| = |y_2| = M$ and $\arg(y_i) = 2\pi\nu/w_s$ where w_s is the sampling frequency in Hertz; M is the magnitude of the root ($M > 1$ for causality) and ν the frequency index, $\nu \in (0, w_s/2)$. The spectrum will have modal frequency in ν , which will be broader as $M \rightarrow \infty$ and narrower as $M \rightarrow 1^+$. Then, given (ν, M, w_s) and with $\omega_0 = \frac{2\pi\nu}{w_s}$ we have $u_1 = 2M^{-1} \cos \omega_0$ and $u_2 = -M^{-2}$.

Two groups of 50 time series each were simulated, with parameters $\nu_1 = 0.21$, $\nu_2 = 0.22$, $M_1 = M_2 = 1.15$, $w_s = 1$ and length $T = 1000$. From the simulated time series, the spectral densities were estimated using a smoothed lag-window estimator with a Parzen window and bandwidth $100/T$. The estimated spectral densities are shown in Fig. 1(a). The functional form of the estimated spectral densities was recovered using a B-Spline basis of degree 3 with 14 equispaced nodes and smoothing parameter $\lambda = 0.000003$ (see e.g. [22], Chap. 3) We want to test the performance of the different algorithms in recovering these two groups, even in the presence of contaminating data. In the absence of contamination we have 100 observations divided into two groups.

We introduce the mixtures of AR(2) processes that will be used in the contamination schemes. Let $Y_t^i, i = 1, 2$ be two AR(2) processes with parameters M_i and $\nu_i, i = 1, 2$. Their mixture is given by $X_t = a_1 Y_t^1 + a_2 Y_t^2 + \epsilon_t$ where $a_i, i = 1, 2$ are the weights and ϵ_t is a white noise process. This mixture creates a signal that combines the oscillatory behavior of the processes $Y_t^i, i = 1, 2$.

Starting from the two groups of 50 AR(2) time series described previously, which are considered as the clean data, we added another 11 time series (around 10% contamination level), generated according to the following schemes:

- (i) AR(2) processes with parameters ν_i chosen randomly with uniform distribution in the interval $(.20, .25)$, denoted $U(.20, .25)$, $M = 1.2$ and $w_s = 1$. The contaminating series have smaller variance than the series in the clusters. See Fig. 1(b).
- (ii) A mixture of two AR(2) processes having parameters $\nu_i = .20$ and $.25$; $M_i = 1.05, 1.1, i = 1, 2$ y $w_s = 1$. See Fig. 1(c).
- (iii) A mixture of two AR(2) processes with random parameters $\nu_1 = U(.19, .22)$ y $\nu_2 = U(.24, .26)$; $M_i = 1.05, 1.1, i = 1, 2$ and $w_s = 1$, See Fig. 1(d).

Figure 1(b), (c) and (d) show the spectral densities for the simulated time series with the three contamination schemes described.

In order to test the performance of the RFC methodology, the simulated process and their estimated spectral densities were used to compare with the

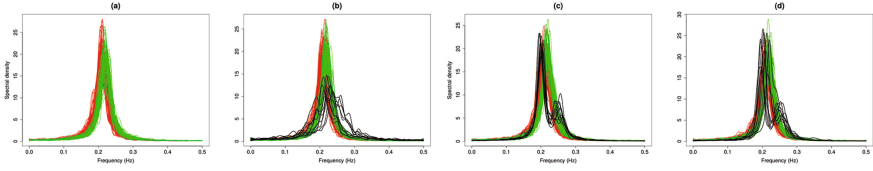


Fig. 1. Spectral density of the simulated time series: (a) No contamination, (b) Contamination type (i), (c) Contamination type (ii) and (d) Contamination type (iii)

results obtained when using the “Funclust” algorithm [17] and hierarchical methods using the total variation distance: “HSMClust” [13] and “TVDClust” [2,3].

It is important to recall that we assume the q_g dimensions in the RFC procedure to be unknown parameters and that the BIC criterion is used to estimate them when applying this algorithm. The results in [24] already show the importance of trimming. Trimming levels $\alpha = 0$ and $\alpha = 0.1$ are used. As regards the constraints, we are assuming $d_1 = d_2 = d$ to simplify the simulation study. Values of $d = 3, 10$ and 10^{10} (i.e., almost unconstrained in this last case) were used. We always return the best solution in terms of the highest BIC value for each combination of all those fixed values of trimming level and constraints. We use 100 random initializations with 20 iterations.

For the “Funclust” method we used the library `Funclustering` [26] in R where the EM algorithm has been initialized with the best solutions out of 20 “short” EM algorithms with only 20 iterations and threshold values of $\varepsilon = 0.001, 0.05, 0.1$ in the Cattell test. For the agglomerative methods we use the library `HSMClust` in R for “HSMClust” and “TVDClust” by means of the algorithm described in [2,3].

Figure 2 shows the results for the simulation study. It is composed of a matrix of graphs, where the rows correspond to the different contamination schemes (uncontaminated in the first row) while the columns correspond to the methodologies tested. The first column corresponds to “Funclust”, the second to “HSMClust”, the third shows the results for the RFC procedure with trimming levels $\alpha = 0$ (untrimmed) and $\alpha = 0.1$ and three constraint levels $d = 3, 10$ and 10^{10} (i.e., almost unconstrained in this last case). The fourth column shows the results corresponding to “TVDClust”. The x -axis corresponds to the threshold applied in the Cattell test for “Funclust”, the procedure in “HSMClust”, the constraint level for RFC and the linkage function for the agglomerative method “TVDClust”, while the y -axis corresponds to the correct classification rate (CCR).

Results show that the hierarchical methods, “HSMClust” and “TVDClust” are better in the absence of contamination, giving very consistent results. However, their performance degrades sharply in the presence of noise. This is not surprising since these procedures were not designed to handle contamination in the sample. The joint use of trimming and constraints in RFC improve the results (CCR) substantially. Results are very good for moderate ($d = 10$) and small ($d = 3$) values of the constraint constants, while for high values the results are

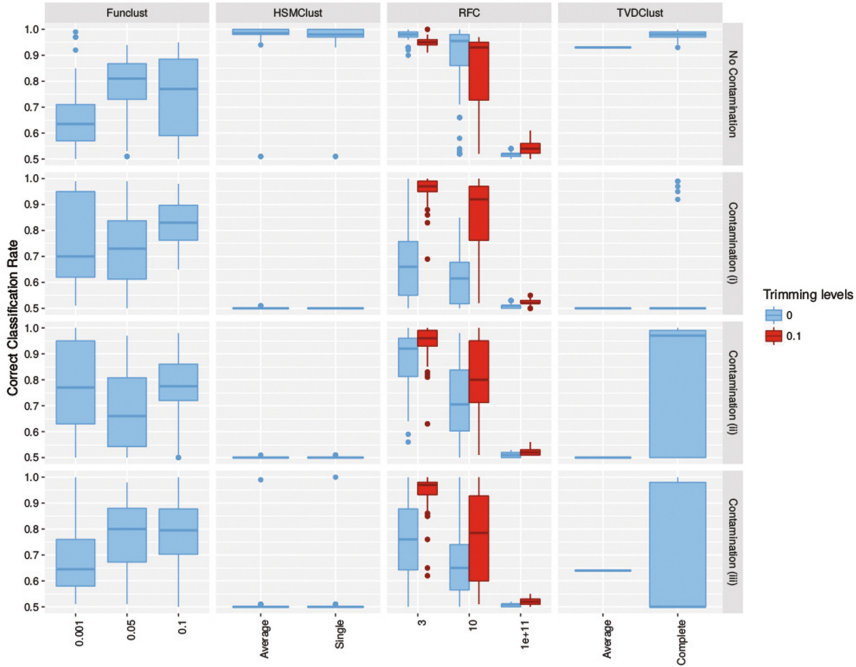


Fig. 2. classification rate (CCR) for the four methods considered, represented in different columns. Rows correspond to the different contamination schemes, starting with no contamination in the first row and following with schemes (i), (ii) and (iii) described in the text. Constraint levels $d_1 = d_2 = 3, 10$ and 10^{10} , trimming levels $\alpha = 0$ and 0.1 were used for the RFC method. Threshold values $\varepsilon = 0.001, 0.05$ and 0.1 were used for the “Cattell” procedure in “Funclust”. Single and average versions were used for “HSMClust” while average and complete linkage functions were used for “TVDClust”.

poor. Very high values for these constants are equivalent to having unconstrained parameters. Trimming turns out to be very useful in all the contaminated cases while the results are not affected by trimming in the uncontaminated case.

In the presence of contamination, the results for “Funclust”, “HSMClust” and “TVDClust” fall below those of RFC when applying the $\alpha = 0.1$ trimming and small/moderate values d_1 and d_2 for the variance parameters.

5 Analysis of Real Data

We now consider wave-height data measured by a buoy located in Waimea Bay, Hawaii, at a water depth of approximately 200 m. This buoy is identified as number 106 (51201 for the National Data Buoy Centre). The data, which corresponds to 72.5 h divided into 30-minute intervals, was collected in June 2004 and has previously been analyzed by [2] where more details can be found.

In [2] the spectrum for each 30-minute interval was estimated and normalized. The TV distance between all spectral densities was used to build a dissimilarity matrix, that was fed into a hierarchical agglomerative clustering algorithm. For more details see [3]. The 145 normalized densities are shown in Fig. 3(a).

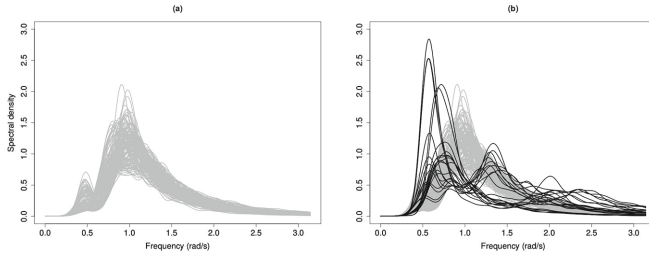


Fig. 3. Spectral densities for the sea wave data after normalization. (a) Original data. (b) Original data plus 22 additional densities in black, considered as noise.

The RFC method was applied to this data set in order to obtain an alternative clustering. The functional form of the data was recovered using B-splines of order 3 with 31 equispaced nodes. We use 100 initializations with 20 iterations each. The constraint level considered was $d_1 = d_2 = 3$, and the trimming level $\alpha = 0.13$. In [2] two different clusterings were obtained, depending on the linkage function used: 4 clusters for the complete linkage and 3 for average. We will only consider the clustering into 4 groups for comparison purposes in what follows.

To compare the two results, the Adjusted Rand Index (ARI) [16] was used. This is an improvement of the original Rand Index [23] and measures the similarity between two partitions of the same set, having value 1 when there is exact coincidence and close to 0 when considering a completely “random” clustering of the data.

One can see that the effect of trimming and constraints is not harmful, even in the absence of contamination. For instance, we can see that the ARI of RFC with $d_1 = d_2 = 3$ and $\alpha = 0.13$ is equal to 0.513 with respect to the “reference” partition, which is obtained when applying [2] with 4 groups. To compute this ARI index we assign all the time series (trimmed and non-trimmed) to clusters by using posterior probabilities from the fitted mixture model that was described in Sect. 3. The two rows in Fig. 4 show the clusters found when using the TVD and RFC, respectively. Even though the groups have differences in membership and size, it is possible to see from the figures that the shape of the functions in the corresponding clusters are very similar and the mean functions are close. The variations are probably due to the different clustering techniques employed, but the similarity in the groups obtained point to consistent results for both methods. Observe that the trimmed curves for the RFC method are different from the rest of the functions in their cluster. For “HMSClust” both versions gave a value of 0.723 for the ARI, higher than that obtained with RFC, while

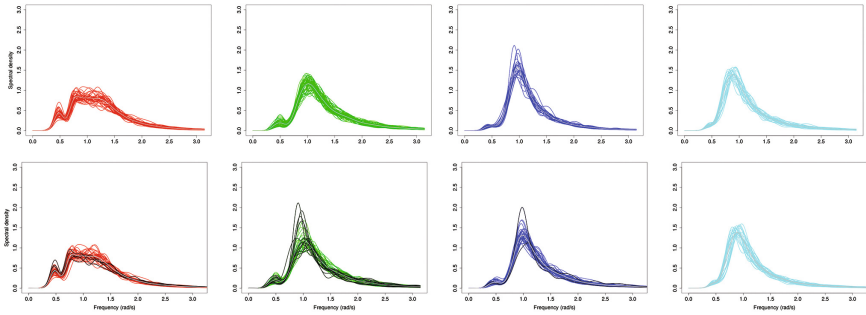


Fig. 4. (Top) Clusters found using the TV distance using the complete linkage function. (Bottom) Clusters found using the RFC method for $K = 4$ with constrains $d_1 = d_2 = 3$ and trimming $\alpha = 0.13$. Each panel corresponds to spectral densities in each cluster, grey lines represent the means and black lines represent the trimmed observations.

for “Funclust”, values were lower, with a maximum of 0.315 with a threshold value of 0.01 or 0.1 in the “Cattell” test.

In order to test the performance of the different methods with real data and in the presence of contamination, 22 time series were added to the sample. These measurements were recorded at the same location and during the same month, but during different days. The corresponding estimated spectral densities are shown in black in Fig. 3(b). Some of these densities are bimodal while others are unimodal but have lower modal frequency than those in the original sample.

The four clustering procedures considered were applied to this contaminated sample and the results were compared using again as “reference” the clustering obtained in [2] with 4 groups applied to the clean data (i.e., before adding the contaminating curves). The ARI was computed by taking only into account the classification of the original (non-contaminating) densities. In the case of the RFC methodology, the assignments based on “posterior” probabilities were considered for the wrongly trimmed observations.

The ARI for the RFC method with $K = 4$ and $d_1 = d_2 = 3$ are equal to 0.167, 0.723 and 0.599 when trimming levels $\alpha = 0$, $\alpha = 0.13$ and $\alpha = 0.2$, respectively, are used. The associated ARI when using Funclust are always below 0.21 for all three Cattell thresholds tested (0.002, 0.05 and 0.1) as this method is not designed to cope with outlying curves. The other methods tested, “average TVD” and “average HMSclust”, have even worse results in this contaminated case reaching ARI values equal to 0 in both cases. Therefore, the best results overall were obtained using RFC with a $\alpha = 0.13$ while the other methods show poor results in the presence of contaminating data.

To reinforce previous claims, Fig. 5 shows in the first row, the partition obtained in [2] with four clusters before adding the contaminating time series, in the second, the results when using RFC with four clusters, $d_1 = d_2 = 3$ and trimming level $\alpha = 13\%$ to the “contaminated” data set. In the third row the results obtained with “TDVclust”, then “HMSclust” and “Funclust”, also in case that

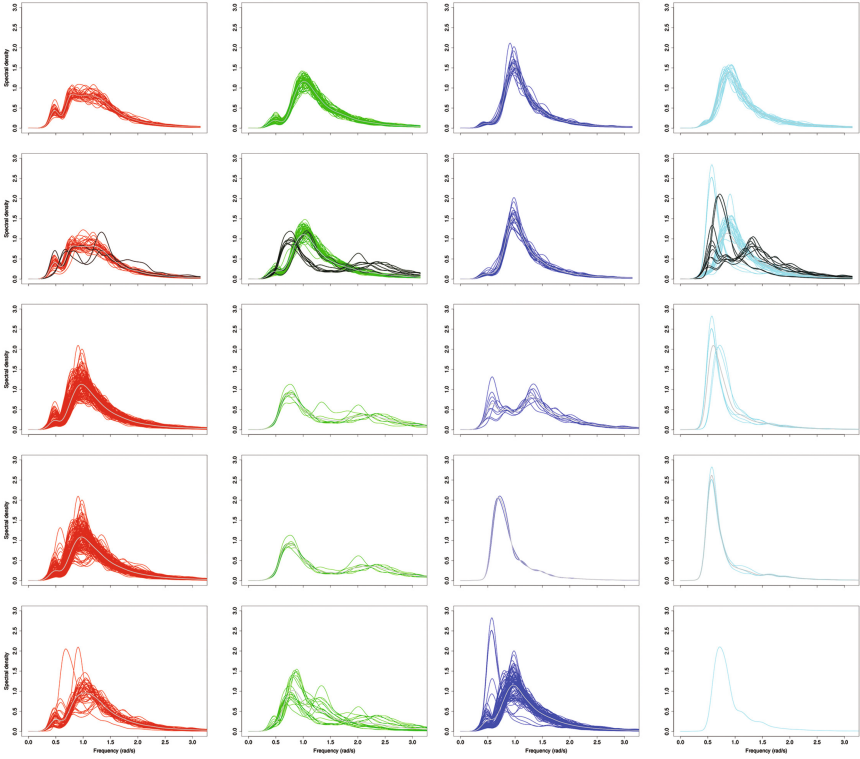


Fig. 5. Clusters found with the different procedures when $K = 4$. Each panel corresponds to a cluster of spectral densities. Gray lines represent the means and black lines represent the trimmed observations. *First row:* Original clusters for the TV distance using the complete link before adding contaminating time series. *Second row:* Clusters for the RFC method for $K = 4$ with constrains $d_1 = d_2 = 3$ and trimming $\alpha = 0.13$. *Third row:* Clusters for the “TVDClust” method with complete linkage. *Fourth row:* Clusters for “HSMClust” method. *Fifth row:* Clusters for “FuncIust”

the contaminating time series were added. Once again, the clusters obtained with RFC differ slightly from those obtained in [2] but, in spite of the presence of contamination, the shape of the spectral densities in the corresponding clusters are very similar and the average densities are very close. The trimmed functions when using level $\alpha = 13\%$ are shown in black in the second row. The last three rows show the poor results obtained with the other three methods. For instance, in the third row, corresponding to “TVDClust”, the original sample is clustered together in a single group in the leftmost panel, while the other three groups only contain contaminating functions that were added as noise. Since $\alpha = 0.13$, 19 curves were trimmed with the RFC procedure, most of which come from the contaminating series that were added. Finally, it is also important to point out that trimming and clustering are performed simultaneously in the RFC approach.

6 Conclusions

A feasible methodology of robust clustering for stationary time series has been proposed and illustrated. The key idea behind the algorithm presented is the use of estimated spectral densities of the time series, that are considered as functional data. A robust model-based algorithm together with the simultaneous use of trimming and constraints is then used to cluster the original time series.

The use of trimming protects the estimation of the parameters against the effect of outlying curves, while the constraints avoid the presence of spurious clusters and improve the performance of the algorithms. Simulations show that the joint use of constraints and trimming tools improves results in the presence of outliers, in comparison to some other procedures for time series and functional data clustering, not designed to work with contamination. The real data example shows that the proposed RFC method for time series clustering has a good performance, with or without the presence of outlying curves. In the presence of contamination, RFC is able to detect almost all the outliers in the data. The trimmed curves often correspond to curves with different characteristics to the rest. We conclude that the proposed robust methodology can be a useful tool to detect contamination and groups in a time series data set simultaneously.

However, this methodology has some limitations. The choice of trimming level α and the choice of the scatter constraints constants d_1 and d_2 , can be subjective and sometimes depend on the final purpose of the cluster analysis. For this reason, we always recommend the use of different values of trimming and constraint, monitoring the effect in the clustering partition of these choices.

Acknowledgements. Data for station 160 were furnished by the Coastal Data Information Program (CDIP), Integrative Oceanographic Division, operated by the Scripps Institution of Oceanography (<http://cdip.ucsd.edu/>). Research by DRG and JO was partially supported by Conacyt, Mexico Proyecto 169175 Análisis Estadístico de Olas Marinas, Fase II. Research by LA G-E and A M-I was partially supported by the Spanish Ministerio de Economía y Competitividad y fondos FEDER, grant MTM2014-56235-C2-1-P, and by Consejería de Educación de la Junta de Castilla y León, grant VA212U13.

References

1. Aghabozorgi, S., Shirkhorshidi, A.S., Wah, T.Y.: Time-series clustering - a decade review. *Inf. Syst.* **53**, 16–38 (2015)
2. Alvarez-Esteban, P.C., Euán, C., Ortega, J.: Statistical analysis of stationary intervals for random waves. In: *Proceedings of International Society of Offshore and Polar Engineering Conference*, vol. 3, pp. 305–311 (2016)
3. Alvarez-Esteban, P.C., Euán, C., Ortega, J.: Time series clustering using the total variation distance with applications in oceanography. *Environmetrics* **27**(6), 355–369 (2016)
4. Bahadori, M.T., Kale, D.C., Fan, Y., Liu, Y.: Functional subspace clustering with application to time series. In: *Proceedings of 32nd International Conference on Machine Learning*, pp. 228–237 (2015)

5. Bouveyron, C., Jacques, J.: Model-based clustering of time series in group-specific functional subspaces. *Adv. Data Anal. Classif.* **5**(4), 281–300 (2011)
6. Caiado, J., Crato, N., Peña, D.: A periodogram-based metric for time series classification. *Comput. Stat. Data Anal.* **50**(10), 2668–2684 (2006)
7. Caiado, J., Crato, N., Peña, N.: Comparison of times series with unequal length in the frequency domain. *Commun. Stat. Simul. Comput.* **38**(3), 527–540 (2009)
8. Caiado, J., Maharaj, E.A., D’Urso, P.: *Time Series Clustering*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. Taylor & Francis, London (2015). Chap. 12
9. Cuesta-Albertos, J.A., Fraiman, R.: Impartial trimmed k -means for functional data. *Comput. Stat. Data Anal.* **51**(10), 4864–4877 (2007)
10. Delaigle, A., Hall, P.: Defining probability density for a distribution of random functions. *Ann. Stat.* **38**(2), 1171–1193 (2010)
11. D’Urso, P., De Giovanni, L., Massari, R.: Time series clustering by a robust autoregressive metric with application to air pollution. *Chemometr. Intell. Lab. Syst.* **141**, 107–124 (2015)
12. D’Urso, P., De Giovanni, L., Massari, R.: GARCH-based robust clustering of time series. *Fuzzy Sets Syst.* **305**, 1–28 (2016)
13. Euán, C., Ombao, H., Ortega, J.: The hierarchical spectral merger algorithm: a new time series clustering procedure. *J. Classif.* (2017, accepted)
14. Fritz, H., García-Escudero, L.A., Mayo-Isicar, A.: A fast algorithm for robust constrained clustering. *Comput. Stat. Data Anal.* **61**, 124–136 (2013)
15. García-Escudero, L.A., Gordaliza, A.: A proposal for robust curve clustering. *J. Classif.* **22**(2), 185–201 (2005)
16. Hubert, L., Arabie, P.: Comparing partitions. *J. Classif.* **2**, 193–218 (1985)
17. Jacques, J., Preda, C.: Funclust: a curves clustering method using functional random variables density approximation. *Neurocomputing* **112**, 164–171 (2013)
18. James, G.M., Sugar, C.A.: Clustering for sparsely sampled functional data. *J. Am. Stat. Assoc.* **98**(462), 397–408 (2003)
19. Liao, T.W.: Clustering of time series data - a survey. *Pattern Recogn.* **38**, 1857–1874 (2005)
20. Maharaj, E.A., D’Urso, P.: Fuzzy clustering of time series in the frequency domain. *Inf. Sci.* **181**(7), 1187–1211 (2011)
21. Montero, P., Vilar, J.: TSclust: an R package for time series clustering. *J. Stat. Softw.* **62**(1), 43 (2014)
22. Ramsay, J.O., Silverman, B.W.: *Functional Data Analysis*. Springer Series in Statistics, 2nd edn. Springer, New York (2005)
23. Rand, W.M.: Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **66**, 846–850 (1971)
24. Rivera-García, D., García-Escudero, L.A., Mayo-Isicar, A., Ortega, J.: Robust clustering for functional data based on trimming and constraints. [arXiv:1701.03267](https://arxiv.org/abs/1701.03267) (2017)
25. Rousseeuw, P.J., Van Driessen, K.: A fast algorithm for the minimum covariance determinant estimator. *Technometrics* **41**, 212–223 (1999)
26. Soueidatt, M.: *Funclustering: A package for functional data clustering*. R package version 1.0.1 (2014)