Clustering Time Series by Linear Dependency

Andrés M. Alonso* Daniel Peña †

Abstract

We present a new way to find clusters in large vectors of time series by using a measure of similarity between two time series, the generalized cross correlation. This measure compares the determinant of the correlation matrix until some lag k of the bivariate vector with those of the two univariate time series. A matrix of similarities among the series based on this measure is used as input of a clustering algorithm. The procedure is automatic, can be applied to large data sets and it is useful to find groups in Dynamic Factor Models. The cluster method is illustrated with some Monte Carlo experiments and a real data example.

Accepted at Statistics and Computing

https://doi.org/10.1007/s11222-018-9830-6

*Department of Statistics and Institute Flores de Lemus, Universidad Carlos III de Madrid, Getafe, Spain. email: andres.alonso@uc3m.es

†Department of Statistics and Institute UC3M-BS of Financial Big Data, Universidad Carlos III de Madrid, Getafe, Spain. email: daniel.pena@uc3m.es

1 Introduction

Most procedures for clustering time series look at the similarity of the elements of a set of times series and build a measure of distance between two series by using their univariate features. Piccolo (1990) proposed a distance measure for classifying ARMA models using the autoregressive representation of the process. Their properties are further studied in Corduas and Piccolo (2008). Xiong and Yeung (2004) applied a model-based approach using mixtures of autoregressive moving average (ARMA) models and the EM algorithm to estimate the parameters. Scotto et al (2011) and D'Urso et al (2017) use the extreme value behaviour for clustering environmental time series. Liao (2005) surveys the time series clustering procedures available from the perspective of machine learning. From the Bayesian approach, Fruhwirth-Schnatter and Kaufmann (2008) build groups by using finite-mixture models estimated by Bayesian Markov Chain Monte Carlo simulation methods. Pamminger and Fruhwirth-Schnatter (2010) developed a model based approach for categorical time series. Also, in the model based approach, Alonso et al. (2006) and Vilar-Fernández et al (2010) cluster time series by using the forecast densities. Specific frequency-domain methods for discrimination and clustering analysis of time series were proposed by Maharaj (2002) and by Caiado et al (2006) or in the fuzzy framework by Maharaj and D'Urso (2011). Pértega and Vilar (2010) compared several parametric and nonparametric approaches. Zhang et al (2011) introduced a two step method in which one-nearest neighbor network is built based on the similarity of time series with the triangle distance, and second the nodes with high degrees are used to cluster. Zhang (2013), Sadahiro and Kobayashi (2014) and Aghabozorgi and Wah (2014) analized methods for high-dimensional time series. Recent surveys of the field can be found in Aghabozorgi et al (2015) and Caiado et al (2015). The R library TSclust implements many of the previous mentioned clustering procedures (see Montero and Vilar, 2014). These methods are useful when we have independent time series and the objective is to cluster them by similarity of their univariate models, in a parametric framework, or by similarity of their periodograms or autocorrelation functions.

For a set of independent realizations of vectors of stationary time series Kakizawa et al (1998) developed measures of disparity between these vectors using the Kullback-Leibler and Chernoff information measures and proposed a spectral approximation to define quasi-distances between the time series. These measures are then used in a hierarchical, or k-means partitioned, cluster algorithm.

In this article we propose a procedure to cluster time series by their linear dependency. Our approach is general and nonparametric, as it does not assume any model for the series, and it is based on the cross-correlation coefficients between them. Golay et al. (2005) and Douzal-Chouakria and Nagabhushan (2007) have proposed using variants of the instantaneous cross-correlation, but their approach take into account the sign of the cross-correlation. Clustering by dependency has recently been analyzed by Ando and Bai (2016, 2017) assuming that the vector of time series is generated by a Dynamic Factor Model

where some factors affect different groups of series. They propose an iterative estimation algorithm to find the clusters based on factor model estimation.

The contributions of this article are as follows. We first justify in Section 2 that clustering by similar linear dependency, taking into account the cross correlations, will produce different results than clustering for similar univariate structure, using the autocorrelations. The main contribution of this article appears in Section 3, that defines a general measure of linear association between two time series, the generalized cross correlation (GCC), which has useful properties for measuring their linear dependency. In particular, it is shown in Section 4 that is equivalent to using a dynamic version of the mutual information between two series. Section 5 derives an objective criterion for the choice of k, the number of lags to be included in building the GCC, proposes a consistent estimate of this measure and explains its use for clustering time series. Section 6 illustrates in a Monte Carlo study that the proposed procedure has a good performance on cross-dependency clustering compared to both some univariate clustering procedures and to the dependency cluster methods proposed by Douzal-Chouakria and Nagabhushan (2007) and Ando and Bai (2016). Finally, we apply our procedure to series of electricity prices in Section 7. Some brief conclusions are presented in Section 8.

2 Univariate versus Bivariate clustering

Suppose two stationary linear processes, y_t and x_t, which follow a vector ARMA(p_b, q_b) model

$$\begin{bmatrix} \phi_{11}(B) & \phi_{12}(B) \\ \phi_{21}(B) & \phi_{22}(B) \end{bmatrix} \begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} \theta_{11}(B) & \theta_{12}(B) \\ \theta_{21}(B) & \theta_{22}(B) \end{bmatrix} \begin{bmatrix} a_{1t} \\ a_{2t} \end{bmatrix}, \tag{1}$$

where a_{1t} and a_{1t} are white noise processes with covariance matrix Σ_a and the polynomials $\phi_{ij}(B)$ and $\theta_{ij}(B)$ are of the form $A_{ii}(B) = 1 - a_{1ii}B - \ldots - a_{rii}B^r$ or $A_{ij}(B) = -a_{1ij}B - \ldots - a_{rij}B^r$. The univariate models for the time series are

$$\alpha(B)x_t = \beta_1(B)a_{1t} + \beta_2(B)a_{2t} = \beta_x(B)u_{1t},$$

and

$$\alpha(B)y_t = \beta_3(B)a_{1t} + \beta_4(B)a_{2t} = \beta_y(B)u_{2t},$$

where $\alpha(B) = (\phi_{11}(B)\phi_{22}(B) - \phi_{12}(B)\phi_{21}(B))$ is a polynomial of maximum order $2p_b$ and $\beta_j(B)$ for $j = 1, \ldots, 4$ are polynomials of maximum order $p_b + q_b$. As the sum of two MA processes that do not have lag cross correlation is another MA process, the univariate models will be ARMA(p_u, q_u) with $p_u \leq 2p_b$, and MA order, $q_u \leq p_b + q_b$ (see Granger and Morris, 1976). These models will be, in general, different in both series. For instance, suppose a $VAR(1)$ with $\phi_{11}(B) = 1, \phi_{12}(B) = 0$, $\phi_{21}(B) = -aB$, $\phi_{22}(B) = 1 - \phi B$. Then x_t is white noise, whereas y_t follows an AR(1), if Σ_a is diagonal, or it follows an ARMA(1,1) if the noises have contemporaneous correlation.

Thus, using a cluster method based on similar univariate properties of the series may classify two series strongly related in different groups, and put together independent series that follow similar models. These procedure will not be useful if we want to find cluster of related series, as it is usually the objective in many applications. A possible strategy for clustering time series could be to use a two step method. First, the series are clustered by their dependency, and, second, these more homogeneous clusters can be broken down into new clusters of similar univariate structure. In this article we will concentrate in this first stage, the second can be carried out by the procedures discussed in the Introduction.

3 The generalized cross correlation measure between two time series

3.1 Definition of the generalized cross correlation measure

We want to define a general measure of linear dependency among two stationary time series, x_t and y_t. Suppose that, without loss of generality, the series are standardized so that $E(x_t) = E(y_t) = 0$ and $E(x_t^2) = E(y_t^2) = 1$. Call $\rho_{xx}(h) = E(x_{t-h}x_t) = \rho_x(h)$ and $\rho_{xy}(h) = E(x_{t-h}y_t) = \rho_{yx}(-h)$. We assume that the two series are not deterministic and given any finite vector $a = (a_1, \ldots, a_k)$ and series $X_{t,k} = (x_t, \ldots, x_{t-k})$, $P(a'X_{t,k} = 0) = 0$.

The measure of dependency we search for, $C(x_t, y_t)$, should verify the following properties: (1) $0 \leq C(x_t,\, y_t) \leq 1$; (2) $C(x_t, y_t) = 1$ if and only if there is an exact linear relationship between the two series; (3) $C(x_t, y_t) = 0$ if, and only if, all the cross correlation coefficients between the two time series are zero; (4) If both series are white noise and $\rho_{xy}(h) = 0$ for $h \neq 0$, then $C(x_t, y_t) = \rho_{xy}^2(0)$.

We can summarize the linear dependency at lag h by the matrix

$$\mathbf{R}(h) = \begin{pmatrix} \rho_x(h) & \rho_{xy}(h) \\ \rho_{yx}(h) & \rho_y(h) \end{pmatrix} \tag{2}$$

and, in the same way, the linear dependency for lags between 0 and k can be summarized by putting together the $(k+1)$ square matrices of dimension two that describe the dependency at each lag, as

$$\mathbf{R}_k = \begin{pmatrix} \mathbf{R}(0) & \mathbf{R}(1) & \ldots & \mathbf{R}(k) \\ \mathbf{R}(-1) & \mathbf{R}(0) & \ldots & \mathbf{R}(k-1) \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{R}(-k+1) & \mathbf{R}(-k+2) & \ldots & \mathbf{R}(1) \\ \mathbf{R}(-k) & \mathbf{R}(-k+1) & \ldots & \mathbf{R}(0) \end{pmatrix}. \tag{3}$$

The matrix $\mathbf{R}_k$ is symmetric non negative definite, and it corresponds to the covariance matrix of the vector stationary process $(x_t, y_t, x_{t-1}, y_{t-1}, \ldots, x_{t-k}, y_{t-k})'$. We can arrange the components of the vector as $Z_{t,2(k+1)} = (Y'_{t,k}, X'_{t,k})'$,

with covariance matrix

$$\mathbf{R}_{yx,k} = \begin{pmatrix} 1 & \rho_y(1) & \dots & \rho_y(k) & \rho_{xy}(0) & \rho_{xy}(1) & \dots & \rho_{xy}(k) \\ \rho_y(1) & 1 & \dots & \rho_y(k-1) & \rho_{xy}(-1) & \rho_{xy}(0) & \dots & \rho_{xy}(k-1) \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \rho_y(k) & \rho_y(k-1) & \dots & 1 & \rho_{xy}(-k) & \rho_{xy}(-k+1) & \dots & \rho_{xy}(0) \\ \rho_{xy}(0) & \rho_{xy}(-1) & \dots & \rho_{xy}(-k) & 1 & \rho_x(1) & \dots & \rho_x(k) \\ \rho_{xy}(1) & \rho_{xy}(0) & \dots & \rho_{xy}(-k+1) & \rho_x(1) & 1 & \dots & \rho_x(k-1) \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \rho_{xy}(k) & \rho_{xy}(k-1) & \dots & \rho_{xy}(0) & \rho_x(k) & \rho_x(k-1) & \dots & 1 \end{pmatrix} \tag{4}$$

$$= \begin{pmatrix} \mathbf{R}_{yy,k} & \mathbf{C}_{xy,k}^T \\ \mathbf{C}_{xy,k} & \mathbf{R}_{xx,k} \end{pmatrix}, \tag{5}$$

where $\mathbf{R}_{xx,k}$ is the $(k+1)$ squared and positive definite covariance matrix of the standardized vector of series $X_{t,k} = (x_t, x_{t-1}, \dots, x_{t-k})'$, $\mathbf{R}_{yy,k}$ corresponds to $Y_{t,k} = (y_t, y_{t-1}, \dots, y_{t-k})'$ and $\mathbf{C}_{xy,k}$ include the cross correlations between both vectors of series. Note that $|\mathbf{R}_{yx,k}| = |\mathbf{R}_k|$. This matrix verifies that (1) $0 \leq |\mathbf{R}_{yx,k}| \leq 1$, with equality to one holding when $\mathbf{R}_{yx,k}$ is diagonal and the two series are both serially uncorrelated and not linearly related; (2) $|\mathbf{R}_{yx,k}| = 0$ when there exists a linear combination $a'Z_t = 0$ so that the series are exactly linearly related.

We will call *total correlation* between the two time series, x_t and y_t, to

$$TC = 1 - |\mathbf{R}_{yx,k}|^{1/2(k+1)} \tag{6}$$

which is a measure of the distance of the two series from a bivariate white noise process, that has $TC = 0$. A similar measure applied to a single time series was proposed as a portmanteau test by Peña and Rodríguez (2002), and it was extended as a multivariate portmanteau test by Mahdi and McLeod (2012). A related statistic based on the determinant of the cross correlation matrix of the residuals of a VARMA model has been proposed as an independence test by Robbins and Fisher (2015). However, $|\mathbf{R}_{yx,k}|$ is not a good measure of the strength of the linear relationship between the two series, because it depends on both the cross correlations and the autocorrelations of both series. First, a large value of this determinant does not imply that the series have a weak relationship or are not linearly related. As

$$|\mathbf{R}_{yx,k}| = |\mathbf{R}_{xx,k}| \left| \mathbf{R}_{yy,k} - \mathbf{C}_{xy,k}\mathbf{R}_{xx,k}^{-1}\mathbf{C}_{xy,k}^T \right|, \tag{7}$$

if $\mathbf{C}_{xy,k} = 0$, then $|\mathbf{R}_{yx,k}| = |\mathbf{R}_{xx,k}|\,|\mathbf{R}_{yy,k}|$ which can be very small when the series have strong autocorrelations. For instance, $|\mathbf{R}_{xx,1}| = 1 - \rho_x^2$ will be very small if the first autocorrelation coefficient is close to one. Second, although $|\mathbf{R}_{yx,k}| = 0$ implies an exact relationship between the two series the opposite is not true: a small value of this determinant does not imply a strong relationship between the series. For instance, if $|\mathbf{R}_{xx,k}|$ is very small, because there is strong autocorrelation, then, by (7), $|\mathbf{R}_{yx,k}|$ will also be small.

These properties of $|\mathbf{R}_{yx,k}|$ suggests the following alternative similarity measure

$$GCC(x_t, y_t) = 1 - \left(\frac{|\mathbf{R}_{yx,k}|}{|\mathbf{R}_{xx,k}|\,|\mathbf{R}_{yy,k}|} \right)^{1/(k+1)} \tag{8}$$

$$= 1 - \frac{\left|\mathbf{R}_{yy,k} - \mathbf{C}_{xy,k}\mathbf{R}_{xx,k}^{-1}\mathbf{C}_{xy,k}^{T}\right|^{1/(k+1)}}{|\mathbf{R}_{yy,k}|^{1/(k+1)}}. \tag{9}$$

that we named *generalized cross correlation measure*, $GCC(x_t, y_t)$ between two time series. The measure GCC verifies: (1) $GCC(x_t, y_t) = GCC(y_t, x_t)$ and the measure is symmetric; (2) $0 \leq GCC(x_t, y_t) \leq 1$ for Fischer's inequality (see Lütkepohl, 1996); (3) $GCC(x_t, y_t) = 1$ if and only if there is a perfect linear dependency among the series and (4) $GCC(x_t, y_t) = 0$ if and only if all the cross correlation coefficients are zero.

To prove (3), as $GCC(x_t, y_t) = 1$ implies $|\mathbf{R}_{yx,k}| = 0$, because the denominator is bounded, and then there is at least a row (column) which is a linear combinations of the others rows (columns). To prove (4), write $GCC(x_t, y_t)$ as

$$GCC(x_t, y_t) = 1 - \left| \begin{pmatrix} \mathbf{R}_{xx,k}^{-1} & \mathbf{0}_{xy,k} \\ \mathbf{0}_{yx,k} & \mathbf{I}_{yy,k} \end{pmatrix} \begin{pmatrix} \mathbf{R}_{xx,k} & \mathbf{C}_{xy,k} \\ \mathbf{C}_{xy,k}^{T} & \mathbf{R}_{yy,k} \end{pmatrix} \begin{pmatrix} \mathbf{I}_{xx,k} & \mathbf{0}_{xy,k} \\ \mathbf{0}_{yx,k} & \mathbf{R}_{yy,k}^{-1} \end{pmatrix} \right|^{1/(k+1)} \tag{10}$$

$$= 1 - \left| \begin{pmatrix} \mathbf{I}_{xx,k} & \mathbf{R}_{xx,k}^{-1}\mathbf{C}_{xy,k}\mathbf{R}_{yy,k}^{-1} \\ \mathbf{C}_{xy,k} & \mathbf{I}_{yy,k} \end{pmatrix} \right|^{1/(k+1)}. \tag{11}$$

By the Hadamard's inequality, the right hand side determinant is smaller or equal to 1 and equality is achieved if and only the matrix is diagonal, that is, if and only if $\mathbf{C}_{xy,k} = \mathbf{0}_{xy,k}$.

Notice that for $k = 0$ the $GCC(x_t, y_t)$ is just the squared correlation coefficient between the two variables. Also, for any k, when both series are white noise and $\rho_{xy}(h) \neq 0$ for some $h \neq 0$ and $\rho_{xy}(j) = 0$ for all $j \neq h$, then $GCC(x_t, y_t) = \rho_{xy}^2(h)$. In general, for $k > 0$, we will show in the next section that the $GCC(x_t, y_t)$ represents the increase in accuracy in prediction of the bivariate model with respect to the univariate models and it can be interpreted as an average squared correlation coefficient when we explain the residuals of an autoregressive fitting of one variable by the values of the other.

3.2 Interpretation of the generalized cross correlation measure

In order to understand better this measure note that we can write (see Peña and Rodríguez, 2002)

$$|\mathbf{R}_{yx,k}| = |\mathbf{R}_{yx,k,-1}| \left(1 - R_{1,2k+1}^2(y_t / y_{t-1}, \ldots, y_{t-k}, X'_{t,k})\right), \tag{12}$$

where $\mathbf{R}_{yx,k,-1}$ is the correlation matrix of the vector $Z_{t,2k+1} = (y_{t-1}, \ldots, y_{t-k}, X'_{t,k})'$, in which we have dropped the first component in $Z_{t,2(k+1)}$, and $R_{1,2k+1}^2$

$(y_t/y_{t-1},\ldots,y_{t-k},x_t,\ldots,x_{t-k})$ is the square of the multiple correlation coefficient in the linear fit of the first component of $Z_{t,2(k+1)}$ using as regressors the remaining $2k+1$ variables. In other words, the notation $(y_t/y_{t-1},\ldots,y_{t-k},x_t,\ldots,$ $x_{t-k})$ denote the regression $\widehat{y}_t=\sum_{j=1}^{k}c_jy_{t-j}+\sum_{j=0}^{k}b_jx_{t-j}$. By recursive use of this expression, we have

$$|\mathbf{R}_{yx,k}|=\prod_{i=1}^{2(k+1)}(1-R_z^2(i,2(k+1)-i)), \tag{13}$$

where $R_z^2(i,2(k+1)-i)$ is the multiple correlation coefficient in the regression of the ith component of the vector $\mathbf{Z}_{t,2(k+1)}$ on the remaining $2(k+1)-i$ variables, and $R^2_{2(k+1),0}=0$. Note that: (1) for $1\leq i\leq k+1$, then $R_z^2(i,2(k+1)-i)$ is the multiple correlation coefficient in the regression $(y_t/y_{t-1},\ldots,y_{t-k+i-1},x_{t+i-1},\ldots,\ x_{t-k+i-1})$ and (2) for $k+2\leq i\leq 2(k+1)$ corresponds to the regression $(x_t/x_{t-1},\ldots,\ x_{t-2(k+1)+i})$. Define $R_x^2(i,(k+1)-i)$ and $R_y^2(i,(k+1)-i)$ in the same way for $i=1,\ldots.k+1$ and note that for $k+2\leq i\leq 2(k+1)$ then $R_z^2(i,2(k+1)-i)=R_x^2(j,(k+1)-j)$ for $j=i-(k+1)$. We have, for each individual series

$$|\mathbf{R}_{xx,k}|=\prod_{i=1}^{k+1}(1-R_x^2(i,k+1-i)), \tag{14}$$

Thus, $GCC(x_t,y_t)$ can be written by (13) and (14) as

$$GCC(x_t,y_t)=1-\left(\frac{\prod_{i=1}^{k+1}(1-R_z^2(i,2(k+1)-i)}{\prod_{i=1}^{k+1}(1-R_y^2(i,k+1-i))}\right)^{1/(k+1)} \tag{15}$$

and this coefficient is the ratio between geometric mean of the residual variability in the regressions $(y_t/y_{t-1},\ldots,y_{t-k+i-1},x_{t+i-1},\ldots,x_{t+i-k-1})$ and $(y_t/y_{t-1},$ $\ldots,y_{t-k+i-1})$. For a given regression we call ESS the explained sum of squares variation, USS the unexplained sum of squares variation and $TSS=ESS+$ USS the total sum of squares. Then

$$\frac{(1-R_z^2(i,2(k+1)-i)}{(1-R_y^2(i,k+1-i))}=\frac{USS(i,2(k+1)-i)}{USS(i,k+1-i)}=1-R^2_{e_{i/X_t}}, \tag{16}$$

where $R^2_{e_{i/X_t}}$ is the squared correlation coefficient when (1) we first fit by Least squares the $AR(k-i+1)$ autoregressive $y_{t-i+1}=\phi_1y_{t-i}+\ldots+\phi_{k-i+1}y_{t-k}$ and compute the residuals, $e^{(i)}_{t-i+1}=y_{t-i+1}-\widehat{\phi}_1y_{t-i}-\ldots\widehat{\phi}_{k-i+1}y_{t-k}$; (2) we regress these residuals $e_t^{(i)}$on the vector $X_{t+i-1,k}$. Note that in these last regressions we are using ith lead values of x_t, the contemporaneous observations, and $k-i$ lags of x_t. Therefore, we can write

$$GCC(x_t,y_t)=1-\left(\prod_{i=1}^{k+1}(1-R^2_{e_{i/X_t}})\right).^{1/(k+1)} \tag{17}$$

An alternative interpretation of $GCC(x_t,y_t)$ can be obtained by using $|\mathbf{R}_k|$ instead of $|\mathbf{R}_{yx,k}|$ in the definition of (9) and applying the same ideas. It is easy to

see that we compare now the ratios of $2(k+1)$ regressions. Half of them are of the form $(x_t/x_{t-1},\ldots,x_{t-k+i-1},y_t,\ldots,\ y_{t+i-k-1})$ versus $(x_t/x_{t-1},\ldots,x_{t-k+i-1})$, where we always use the same number of lags for both variables and we always include in the regression the contemporaneous value of the other variable. The other half are of the form $(y_t/y_{t-1},\ldots,y_{t-k+i-1},x_{t-1},\ldots,x_{t+i-k-1})$ versus $(y_t/y_{t-1},\ldots,y_{t-k+i-1})$, and are similar to the previous ones but now the contemporaneous value of the regressors variable is not included.

The name generalized cross correlation has been used before in the signal processing literature to describe an algorithm of maximum likelihood estimation of the time delay between two signals (Knapp and Carter, 1976). For this reason we have used the name generalized cross correlation measure to differentiate it from the algorithm with the same name but with different objective. The chosen name relates it to the generalized variance (Anderson, 1984), based on the determinant of the covariance matrix. This measure is also related to the effective dependence between two random variables (Peña and Rodríguez, 2003). The likelihood ratio test to check that the vector of standardized normal variables $\mathbf{X}'_k$ and $\mathbf{Y}'_k$ are independent assuming multivariate normality is $T\log(|\widehat{\mathbf{R}}_{xx,k}||\widehat{\mathbf{R}}_{yy,k}|/|\widehat{\mathbf{R}}_{yx,k}|)$ (see, for instance, Anderson, 1984), where $\widehat{\mathbf{R}}_{yx,k}$ is the estimated obtained by using the sample autocorrelation and cross correlation. This is a sufficient statistic for checking for independence under normality. Thus, the generalized cross correlation GCC includes all the relevant information under normality about the linear dependency among the series.

4 The Generalized Cross Correlation as Dynamic Mutual Information

Given two continuous random variables, x and y, the mutual information, or mean information in one about the other is a measure of their joint dependency (see Kullback, 1968) given by

$$I(x,y)=\int\int \log\frac{f(x,y)}{f(x)f(y)}f(x,y)dxdy,$$

where $f(z)$ is the density function of z. For univariate normal random variables it is easy to see that

$$I(x,y)=-\frac{1}{2}\log(1-\rho_{xy}^2), \tag{18}$$

where ρ_{xy} is the correlation coefficient between them. This definition can be extended in a direct way to vectors of random variables (X,Y) of dimensions p and q and we will use the form

$$I(X,Y)=\int_{R^p}\int_{R^q}\log\frac{f(Y/X)}{f(Y)}f(X,Y)dXdY \tag{19}$$

and if $Y = y$ is a univariate variable and we assume joint multivariate normality for all we can generalize (18) to

$$I(y, X) = -\frac{1}{2}\log(1 - R^2_{y/X}), \tag{20}$$

where $R_{y/X}$ is the multiple correlation coefficient in the regression of y on X.

Suppose now that we take $X_t = (x_t, \ldots, x_{t-k})$ and $Y_t = (y_t, \ldots, y_{t-k})$, where y_t and x_t are stationary time series. Then, we define the dynamic mutual information between the two series as

$$I_D(X_t, Y_t) = \int_{R^k}\int_{R^k} \log \frac{f(y_t/y_{t-1}, \ldots, y_{t-k}, X_t)\ldots f(y_{t-k}/X_t)}{f(y_t/y_{t-1}, \ldots, y_{t-k})\ldots f(y_{t-k})} f(X_t, Y_t),$$

where $I_D(X_t, Y_t)$ has the usual properties of $I(X, Y)$. Assuming multivariate normality for the joint distribution $f(X_t, Y_t)$, and by (20), we can integrate each term in this equation by

$$\int_{R^k}\int_{R^k} \log \frac{f(y_{t-h}/y_{t-h-1}, \ldots, y_{t-k}, X_t)}{f(y_{t-h}/y_{t-h-1}, \ldots, y_{t-k})} f(X_t, Y_t) = -\frac{1}{2}\log(1 - R^2_{e_h/X_t}),$$

where $R^2_{e_h/X_t}$ is the squared correlation coefficient in the regression of the variable $e_{t,h} = y_{t-h} - E(y_{t-h}/y_{t-h-1}, \ldots, y_{t-k})$ on X_t. By recursive application of this result we conclude that

$$I_D(X_t, Y_t) = -\frac{1}{2}\sum_{i=0}^{k} \log(1 - R^2_{e_h/X_t}) = -\frac{k+1}{2}\log(1 - GCC(x_t, y_t)).$$

Therefore in the Gaussian case the generalized cross correlation is a monotonic transformation of the dynamic mutual information between the two series.

5 Clustering time series with the Generalized Cross Correlations

Given a sample of N stationary time series, in order to apply a cluster procedure based on the GCC we need to: (i) decide about the value of k, the number of lags to be used; (ii) estimate the GCC from the data and build a dissimilarity matrix of the series; (iii) define how to use this matrix to cluster the series. We will analyze these three problems below.

5.1 Selecting the number of lags k

In some problems we have prior information about the relevant lags to be used. For instance, suppose we have daily time series and we expect weakly and yearly seasonality. Then, assuming that the transformation $n_t = \nabla_7\nabla_{365}x_t$, where $\nabla_a x_t = x_t - x_{t-a}$, leads to a set of stationary time series, we expect non zero autocorrelations at the first lag and also at lags 7th and 365th, as well as in a

window h around each of these lags for the interaction between the regular and seasonal part around the seasonal autocorrelation coefficients. That is, we may include in k a set of lag values as $(1, \ldots, h)$, $(7-h, \ldots, 7+h)$, $(365-h, \ldots, 365+h)$ for some small value of h.

However, in many applications we do not have prior information about the number of relevant lags k to be used in computing the GCC. Then, we may think in this value as a parameter that indicates the largest lag that contains additional information about both the cross correlations and the autocorrelations of the two series. For a univariate series that follows an $ARMA(p, q)$ model this lag is the number of autocorrelations coefficients needed to obtain consistent estimates of the parameters of the model, that is $p + q$. Given the values of these autocorrelations the rest are non informative, and we will call the univariate memory of a linear $ARMA(p, q)$ time series to $k_u = p + q$.

For a set of N stationary linear time series that follow univariate ARMA(p_i, q_i) models, we define the univariate memory of the system as

$$UM_s = \max_{1 \leq i \leq N} (p_i + q_i). \tag{21}$$

In the same way, the bivariate memory of a pair of time series is the largest lag of autocorrelations and cross-correlations coefficients with additional information about the process. It coincides with the largest lag of the cross correlations matrices needed to obtained consistent estimates of the parameters of the bivariate $VARMA(p_b, q_b)$ model, $k_b = (p_b + q_b)$. As it has been shown in Section 2, the univariate models will be ARMA$(p_u \leq 2p_b, q_u \leq (p_b + q_b))$, where the exact order depend on the cancellation of roots between the AR and MA parts. Let us call c the maximum number of roots cancelled in the univariate models, and assume that $p_b \geq q_b$ so that $c \leq p_b$. Then, the univariate models have maximum order $p_u = 2p_b - c, q_u = (p_b + q_b) - c$, and the univariate memory of the system formed by the two series will be $3p_b + q_b - 2c \geq p_b + q_b$. Thus, the memory of a bivariate system where $p_b \geq q_b$ cannot be larger than the memory of the univariate series.

As in most real time series $p_u \geq q_u$, which implies $p_b \geq q_b$, we expect that the maximum univariate memory of the time series will be an upper bound of the bivariate memory. We define the bivariate memory of the system as the maximum of the bivariate memory in all possible pairs, and

$$BM_2 = \max_{1 \leq i < j \leq N} (p_{ij} + q_{ij}), \tag{22}$$

where $H = N(N-1)/2$ is the number of pairs in the system.

This analysis suggests a feasible way to obtain an upper bound, $k^u \geq k$, for the bivariate memory k of the system of N time series. We can fit $AR(p)$ processes to all the univariate time series, select the order by the BIC criterion, and take $k^u = \max_{1 \leq i \leq N}(p_i)$.

A lower bound for the bivariate system memory, $k_u \leq k$, can be computed by assuming that the set of time series has been generated by a Dynamic Factor Model, $\mathbf{Z}_t = \mathbf{P}\mathbf{f}_t + \mathbf{n}_t$, where $\mathbf{Z}_t$ is a multivariate vector of dimension N, the

matrix $\mathbf{P}$ is $N \times r$ and $\mathbf{f}_t$ is a vector of factors of dimension r and $\mathbf{n}_t$ has some idiosyncratic autocorrelation structure. Note that this is not an important restriction as large sets of time series can always be represented by Generalized Dynamic Factor models (see Hallin and Lippi, 2013). Assuming a contemporaneous relationship between the series and the factors, it is well known that they can be estimated by the common eigenvectors of the lag covariance matrices of $\mathbf{Z}_t$, (see, for instance, Peña and Box, 1987) and the number of factors can be obtained by the test proposed by Lam and Yao (2012). Thus, we can compute the factors, fit an $AR(p)$ model to each of the factor series selecting the order p_i by the BIC criterion and take $k_l = \max(p_i)$. Then k_l is expected to be a lower bound for the true value. First, note that as $z_{it} = \chi_{it} + n_{it}$, where $\chi_{it} = \sum_{j=1}^{r} p_{ij} f_{jt}$ is the common part, if χ_{it} follows an ARMA(a_i, b_i) model, the observed series z_{it} are expected to be ARMA of higher order, depending on the model followed by n_{it}. Although some near cancellation of roots are always expected in these cases in many of the series, the maximum order for the observed series z_{it} is expected to be larger than the maximum order for the common parts, and, therefore, the estimation of the order using only the models for the factors will be a lower bound for the true system order.

If these two estimates lead to the same value of $k = k^u = k_l$, we stop. Otherwise, we check values of k in the interval (k_l, k^u) assuming that the two series $\mathbf{z}_t = (x_t, y_t)'$ are related by a vector VAR(p) model, where $k_l \leq p \leq k^u$. We use VAR models to avoid the problem of cancellation of operators in the bivariate systems that makes the estimation slower and more complicated. We fit models with orders between the bounds and select the order by the BIC criterion. Calling $k_{xy} = p$ the memory order of the pair of time series, the bivariate memory system is

$$k = \max_{1 \leq i < j \leq N} (k_{ij}). \tag{23}$$

However, this approach requires to fit $H = N(N-1)/2$ bivariate VAR models for each order p. As a VAR(k) model $\mathbf{\Phi}_k(\mathbf{B})\mathbf{z}_t = \mathbf{a}_t$ with $\mathbf{\Sigma}_a = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{'}$ where $\mathbf{P}^{'}\mathbf{P} = \mathbf{I}$ and $\mathbf{\Lambda}$ is diagonal can be written in the structural form $\mathbf{P}^{'}\mathbf{\Phi}_k(\mathbf{B})\mathbf{z}_t = \epsilon_t$, with $\mathbf{\Sigma}_\epsilon = \mathbf{\Lambda}$, a faster approach to estimate this model is by fitting the regressions

$$\widehat{z}_{1,t} = \sum_{j=1}^{k} c_j z_{1,t-j} + \sum_{j=0}^{k} b_j z_{2,t-j} \tag{24}$$

for $k_l \leq k \leq k^u$, and choose the value k^* that minimizes the BIC criterion. This procedure provides consistent estimates of the parameters (see section 10.1 of Hamilton, 1994) and, therefore, of the value of k. This equation could also be estimated by Lasso, (Tibshirani, 1996) searching for a sparse solution, but this will increase the computational cost without clear advantages. Note that for each pair two values of k are obtained for the two dependent variables in the regression and the largest is a consistent estimate of the memory order for this pair. If the set of series is very large, we can take samples of pairs of series to estimate the bivariate memory of the system.

5.2 Estimating the generalized cross correlation

The generalized cross correlation for each pair of time series will be estimated by the sample correlation matrices. The almost sure consistency of these matrices holds for second–order stationary and ergodic processes under mild conditions (see, for instance, Theorem 6 in Chapter IV of Hannan, 1970). Since $\widehat{GCC}$ is a continuous function of $\widehat{\boldsymbol{R}}_{\boldsymbol{\mathcal{X}}(i)}$, $\widehat{\boldsymbol{R}}_{\boldsymbol{\mathcal{X}}(j)}$ and $\widehat{\boldsymbol{R}}_{\boldsymbol{\mathcal{X}}(i,j)}$, a Slutsky's theorem argument implies the consistency of this estimator (see, for instance, Theorems 18.8 and 18.10 of Davidson, 1994).

The estimators $\widehat{GCC}(X_i, X_j)$, will be obtained for all pairs (i, j) with $i \neq j$ to construct the following $N \times N$ dissimilarity matrix

$$\boldsymbol{DM}_{\widehat{GCC}} = \begin{pmatrix} 0 & 1-\widehat{GCC}(X_1,X_2) & \dots & 1-\widehat{GCC}(X_1,X_N) \\ 1-\widehat{GCC}(X_2,X_1) & 0 & \dots & 1-\widehat{GCC}(X_2,X_N) \\ \vdots & \vdots & \ddots & \vdots \\ 1-\widehat{GCC}(X_N,X_1) & 1-\widehat{GCC}(X_N,X_2) & \dots & 0 \end{pmatrix}, \tag{25}$$

where the elements of this matrix are defined as $1 - \widehat{GCC}(X_i, X_j)$ in order to have small dissimilarity when the series are highly dependent and high values of dissimilarity when the series are independent.

5.3 Selecting a cluster procedure

The dissimilarity matrix (25) can be used in any cluster procedure which requires this kind of input. If the number of series is not very large we can apply hierarchical clustering with single linkage (also known as nearest neighbor clustering) since it allows us to find some interesting dependence structures. For instance, the chained structure, that is, a structure where series X_i is related to series X_{i+1} but it is independent to series X_{i+2} for $i \geq 1$. Figure 1(a) illustrates this situation, that is, the series 1 - 10 have a chained dependence structure. For simplicity, lets assume that we have three series having chained dependence structure. Once the closest pair of series is selected, then the remaining series will be close to one of the series in the pair but far away from the other series. However, the distance (dissimilarity) calculated by single linkage between the remaining series and the pair is small and this series will be close to the pair of series. Of course, other linkage schemes can be used depending on the dependence structure more likely for the data or we may try different approaches and select the one that provides the most interesting solution. In some applications, we may want to detect only highly cross-correlated groups of time series and, in that case, complete linkage could be the appropriate choice.

When the number of series is large we can apply multidimensional scaling to the $N \times N$ matrix (25) and obtain a new $N \times p$ matrix where p is the number of principal coordinates selected as variables, and usually p is much smaller than N. Then we can input these matrix in a cluster algorithm based on this sort of input.

6 Some Monte Carlo experiments on clustering time series

In this section, we develop several Monte Carlo experiments to evaluate the performance of the proposed similarity measures. In subsection 6.1 we show the results for scenarios where the series are cross-dependent, and in subsection 6.2 we present the results for scenarios where the series follow factorial models as in Ando and Bai (2016).

In the comparison we have also introduced as a reference three univariate measures based on autocorrelations as the QAC (Quantile autocovariances, Lafuente-Rego and Vilar, 2015), the distance between sample autocorrelation coefficients, SAC, and the distance between partial autocorrelation coefficients, PAC. It should be notice that these methods provide a hard partition of the set of time series, but soft (fuzzy) versions are available at D'Urso and Maharaj (2009) and Vilar et al (2018). They are compared with two bivariate measures $TC(x_t, y_t)$ defined in (6), $GCC(x_t, y_t)$ given in (9), the Douzal-Chouakria and Nagabhushan (2007) dissimilarity measure (denoted by DCN) and the Ando and Bai (2016) clustering procedure (denoted by ABC). The series are clustered for each measure by a hierarchical clustering algorithm with the single, complete and average linkages.

6.1 Cross-dependent scenarios

In this section, we consider groups of dependent series and the main goal is to cluster them by their cross-dependence. We consider a set of fifteen time series generated by the model $x_{i,t} = \phi_i x_{i,t-1} + \epsilon_{i,t}$ where for $i = 1, 2, \ldots, 5$ we have $\phi_i = 0.9$, and for $i = 6, 7, \ldots, 15$, we have $\phi_i = 0.2$. Thus, from the univariate structure we have two clear groups. We introduce the cross dependency through the innovations $\epsilon_{i,t}$, which are Gaussian white noise variables but with a dependence structure $\rho(i, j) = E(\epsilon_{i,t}, \epsilon_{j,t})$ which depends on the Scenario. Five scenarios are defined by indicating the non null cross-correlations. All the other possible cross correlations are assumed to be zero.

1. Scenario S.1: $\rho(i, i+1) = .5$ for $i = 1, \ldots, 9$.
2. Scenario S.2: $\rho(i, i+1) = .5$ for $i = 1, \ldots, 9$, and $\rho(i, i+1) = .5$ for $i = 11, \ldots, 14$.
3. Scenario S.3: $\rho(i, j) = .9$ for $i = 1, \ldots, 9$ and $j = i+1, \ldots, 10$.
4. Scenario S.4: $\rho(i, j) = .9$ for $i = 1, \ldots, 9$ and $j = i+1, \ldots, 10$, and $\rho(i, i+1) = 0.5$ for $i = 11, \ldots, 14$.
5. Scenario S.5: $\rho(i, j) = .9$ for $i = 1, \ldots, 9$ and $j = i+1, \ldots, 10$, and for $i = 11, \ldots, 14$ and $j = i+1, \ldots, 15$.

Figure 1 illustrates these five scenarios, where the series are represented as points in an ellipse. A line connecting two points means that there exits non

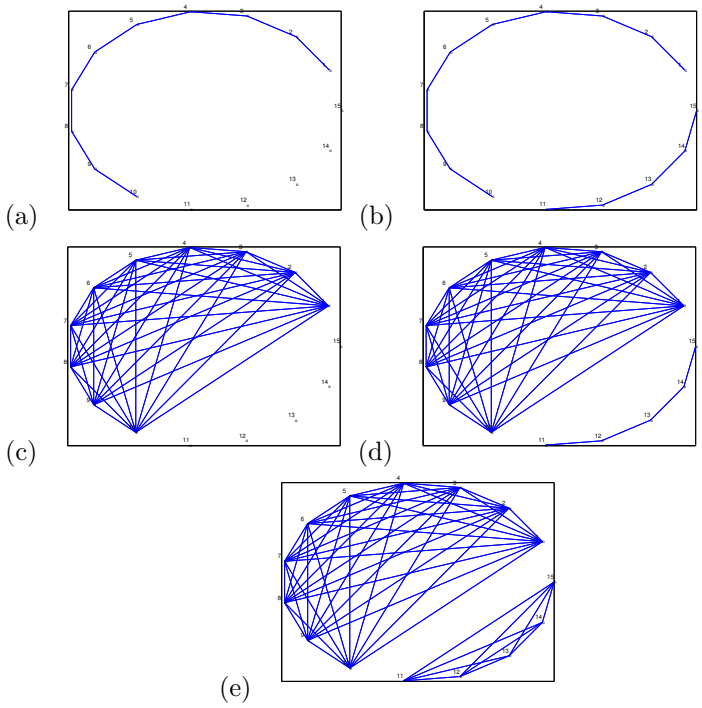

Figure 1: Dependence structure representation. The series are represented as points in the ellipse. A line connecting two points means that there exits non null cross correlation between these two series.

null cross correlation between these two series. For instance, in scenario S.1, represented in Figure 1(a), the first ten series are cross correlated whereas the last five are independent. The Figure shows that scenarios S.2, S.4 and S.5 have two groups of dependent time series, whereas scenarios S.1 and S.3 have one group of dependent time series and five independent time series. A clustering for dependency should lead then to a two-cluster solution for scenarios S.2, S.4 and S.5 and a six-clusters solution for scenarios S.1 and S.3.

For the comparison of the clustering results, we will consider three measures: (1) the adjusted Rand index, ARI, which is based on counting pairs (see Hubert and Arabie, 1985); (2) $\mathcal{F}$-measure which is based on sets overlaps (see Larsen and Aone, 1999) and (3) the variation of information, VI, which is based on mutual information (see, Meilă, 2007). The results obtained with the three measures are similar and we will report here those obtained by using the ARI measure. The results obtained with the $\mathcal{F}$-measure and VI are available upon request to the authors. These measures assume known the "true clusters". It should be notice that there is no generally accepted definition of what the "true clusters" are and this depends on the requirements of the situation, see, e.g., Hennig (2015). In our simulation experiments, we consider the groups of series

Table 1: Clustering performance evaluation (Adjusted Rand Index) at scenarios S.1 – S.5, $T = 100$.

Method	Scenario S.1	Scenario S.2	Scenario S.3	Scenario S.4	Scenario S.5
QAC single	0.025	0.044	0.217	0.045	0.045
SAC single	0.010	0.045	0.187	0.045	0.045
PAC single	-0.011	0.045	0.194	0.045	0.045
DCN single	0.692	0.501	**1.000**	0.973	**1.000**
TC single	0.253	0.038	**1.000**	0.132	0.992
GCC single	**0.945**	**0.894**	**1.000**	**0.988**	**1.000**
QAC complete	0.040	0.045	0.185	0.045	0.046
SAC complete	0.033	0.045	0.175	0.045	0.045
PAC complete	0.026	0.045	0.177	0.045	0.045
DCN complete	0.054	0.003	0.998	0.841	**1.000**
TC complete	0.312	0.176	**1.000**	0.592	**1.000**
GCC complete	0.084	0.103	**1.000**	0.676	**1.000**
QAC average	0.044	0.045	0.218	0.044	0.046
SAC average	0.031	0.045	0.188	0.045	0.045
PAC average	0.018	0.045	0.192	0.045	0.045
DCN average	0.058	0.043	0.998	0.652	**1.000**
TC average	0.612	0.095	**1.000**	0.204	**1.000**
GCC average	0.371	0.614	**1.000**	0.951	**1.000**
PAM - TC	0.553	0.035	0.386	0.059	**1.000**
PAM - GCC	0.092	0.111	0.999	0.712	**1.000**
ABC	-	0.002	-	0.228	0.555

that have some linear dependency as "true clusters", that is, the series that have non null cross correlation.

In Tables 1 and 2, we report the means of the ARI measure from 1000 replicates for these five scenarios when $T = 100$ and 200, respectively. Seven clustering measures are compared: The first three, QAC, SAC and PAC only use univariate information whereas the last four, DCN, TC, GCC and ABC use the bivariate dependency between the series. The number of lags, k, for the methods SAC, PAC, TC and GCC was selected using the procedure described in Section 5. In Tables 3 and 4, we report the frequencies of the selected k as well as the percentage of times where $k^u = k_l$. In method QAC, we use the tuning parameters suggested in Lafuente-Rego and Vilar, (2015). In method DCN, we use the tuning parameter suggested in Montero and Vilar (2014). For method ABC, we assume that each group can be modelled by an unifactorial model. Of course, these scenarios are strongly challenging for Ando and Bai approach since the number of series is small and the procedure relies on a k-means algorithm. In particular, k-means cannot find six clusters in scenarios S.1 and S.3, and provides solutions with empty clusters.

The results of the three univariate methods are very similar across scenarios

Table 2: Clustering performance evaluation (Adjusted Rand Index) at scenarios S.1 – S.5, $T = 200$.

Method	Scenario S.1	Scenario S.2	Scenario S.3	Scenario S.4	Scenario S.5
QAC single	0.024	0.045	0.205	0.045	0.045
SAC single	0.026	0.045	0.189	0.045	0.045
PAC single	0.011	0.045	0.190	0.045	0.045
DCN single	0.972	0.951	**1.000**	0.999	**1.000**
TC single	0.257	0.034	**1.000**	0.132	0.996
GCC single	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
QAC complete	0.043	0.045	0.177	0.045	0.045
SAC complete	0.043	0.045	0.171	0.045	0.045
PAC complete	0.040	0.045	0.171	0.045	0.045
DCN complete	0.061	0.017	**1.000**	0.830	**1.000**
TC complete	0.311	0.146	**1.000**	0.560	**1.000**
GCC complete	0.090	0.108	**1.000**	0.682	**1.000**
QAC average	0.046	0.045	0.207	0.045	0.045
SAC average	0.042	0.045	0.184	0.045	0.045
PAC average	0.036	0.045	0.187	0.045	0.045
DCN average	0.065	0.042	**1.000**	0.649	**1.000**
TC average	0.810	0.124	**1.000**	0.208	**1.000**
GCC average	0.502	0.921	**1.000**	0.998	**1.000**
PAM - TC	0.681	0.013	0.382	0.045	**1.000**
PAM - GCC	0.083	0.108	**1.000**	0.713	**1.000**
ABC	-	0.000	-	0.167	0.511

Table 3: Frequency of selected k at scenarios S.1 – S.5, $T = 100$.

Selected k	Scenario S.1	Scenario S.2	Scenario S.3	Scenario S.4	Scenario S.5
1	59.00%	60.90%	70.30%	72.20%	74.50%
2	30.30%	28.90%	21.50%	19.60%	19.40%
3	7.20%	7.90%	5.60%	6.10%	4.00%
4	2.70%	1.60%	2.30%	1.50%	1.70%
5	0.80%	0.70%	0.30%	0.60%	0.40%
$k^u = k_l$	59.80%	60.90%	69.60%	68.50%	73.90%

Table 4: Frequency of selected k at scenarios S.1 – S.5, $T = 100$.

Selected k	Scenario S.1	Scenario S.2	Scenario S.3	Scenario S.4	Scenario S.5
1	71.10%	69.00%	79.20%	79.50%	81.40%
2	24.10%	25.90%	16.90%	16.90%	14.70%
3	3.70%	3.90%	3.30%	3.00%	2.90%
4	0.80%	1.10%	0.60%	0.50%	0.70%
5	0.30%	0.10%	0.00%	0.10%	0.30%
$k^u = k_l$	69.80%	68.60%	69.90%	67.50%	71.50%

and linkage, as expected. Also, they do not improve with the sample size. This is an expectable result since the univariate methods are not designed for cross-dependency clustering.

The selected number of lags, k, was 1 or 2 in around 90% of the replicates. The percentage of times where $k^u = k_l$ was in the range 59.80% – 73.90%, which produces a low computational cost due the estimation of regressions (24).

The multivariate measures, TC and GCC, have similar performance at the scenarios S.3 and S.5 where there one or two strongly related groups of series. But, the measure TC fails to find the clusters at scenarios S.1, S.2 and S.4 where there is a chained dependence structure. Figure 2 shows an example of the dendrograms obtained using measures TC and GCC for the first scenario. The groups or the independent series at scenario S.1 are clearly distinguishable in the dendrogram based on GCC. But, we observe that TC is not able to distinguish between the groups of series 6–10 and the group of series 11–15 at scenario S.1. Similar graphs are obtained for the other scenarios where there is a chained dependence, S.2 and S.4. The measure DCN obtains better results than TC at scenarios S.1, S.2 and S.4 but it is improved by GCC. It should be noticed that DCN takes into account the sign of the cross-correlation, that is, it consider that two positive correlated time series are closer than two negative correlated time series. This feature does not have impact in scenarios S.1 – S.5 since all non-null cross-correlation are positive but it will determine the behavior of DCN in the factor model scenarios at subsection 6.2.

Also, we observe that complete and average linkages fail to find the clusters in scenarios S.1 and S.2. The complete linkage also fails in scenario S.4. This situation is due to the chain structure of some of the clusters in scenarios S.1, S.2 and S.4. For instance, in the scenario S.1, once the first pair of series is considered as a cluster in the hierarchy then the remaining series will be far away from, at least, one series in this pair. Thus, the farthest neighbour will be far away from this pair. The same argument applies to average linkage although we observe that average linkage is better than complete. In the case of partitioning around medoids procedures, PAM-TC and PAM-GCC, we observe a similar behaviour to average linkage. It should be noticed that there is not a clear medoid in a chained dependence structure.

The proposed measure, the generalized cross-correlation, GCC, with single linkage identifies the related series as well as the independent series and as expected, it improves with the sample size. At scenario S.5, the ABC have a reasonable result since the strong dependence can be assimilated to a factor model. However, it is outperformed by GCC.

6.2 Factor Model scenarios

In this section, we consider three scenarios proposed by Ando and Bai (2016). These scenarios use a dynamic factor model (DFM) with grouped factor structure as data generating process. They consider three groups, each one with three group-specific factors, $r_1 = r_2 = r_3 = 3$, and the same number of elements $N_1 = N_2 = N_3$. In their simulation study, N was 300 and 600, and T was 100

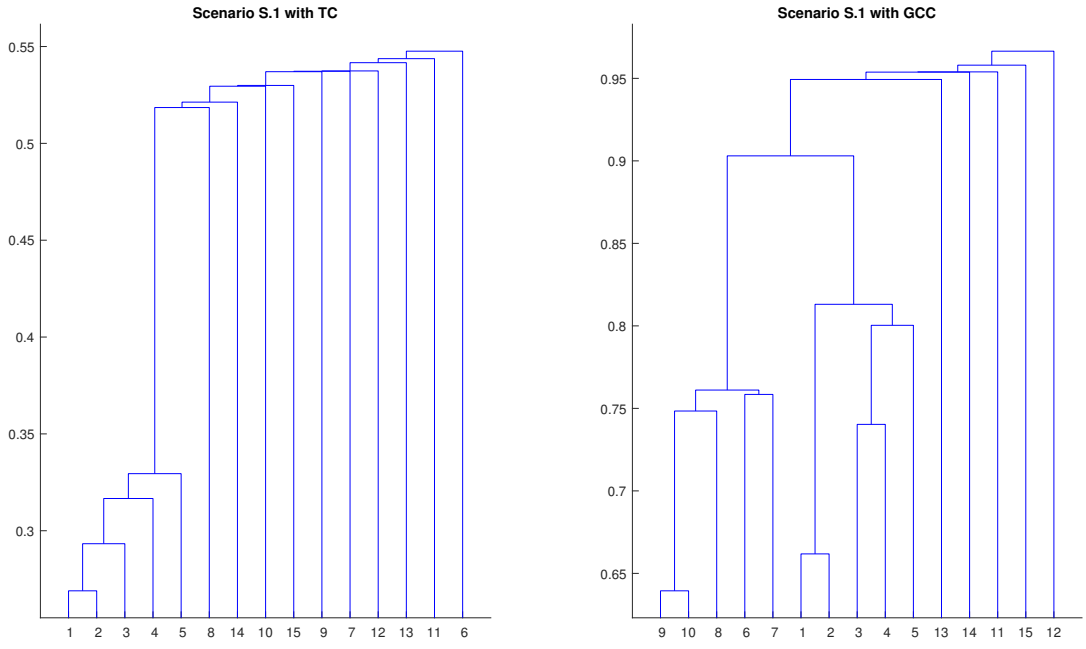

Figure 2: Example of dendrograms (using single linkage) for scenario S.1 obtained with TC and GCC measures.

and 200. The DFM can be expressed as

$$y_{i,t} = \boldsymbol{x}'_{it}\boldsymbol{\beta} + \boldsymbol{f}'_{g_i,t}\boldsymbol{\lambda}_{g_i,i} + \varepsilon_{i,t}, \quad i = 1, 2, \ldots, N, \; t = 1, 2, \ldots, T, \tag{26}$$

where $G = \{g_1, g_2, g_3\}$ denotes the group membership, N_j is the number of elements of group j, $j = \{1, 2, 3\}$, $\boldsymbol{x}_{it}$ is a $p \times 1$ vector of observable variables and $\boldsymbol{f}_{g_i,t}$ is an $r_j \times 1$ vector of unobservable group-specific factors. The r_j dimensional factor of group j is a vector of Gaussian random variables having mean equal to j and unit variances, and the factors loading follows a $N(0, j)$. The vector of observed variables, $\boldsymbol{x}_i$, has dimension 80×1 but only the first three variables are relevant since the parameter $\boldsymbol{\beta}$ was set as $\boldsymbol{\beta} = (1, 2, 3, 0, 0, \ldots, 0)'$.

In model (26), the $p \times 1$ vector of coefficients, $\boldsymbol{\beta}$, is assumed constant across group. Although Ando and Bai (2016) also consider group dependent coefficients the case of constant $\boldsymbol{\beta}$ is a more challenging case when we are interested in finding groups. In this paper, we concentrate our attention on this case.

The three scenarios assume that time series are obtained by model (26), but differ in the properties of the errors, $\varepsilon_{i,t}$:

1. Scenario F.1: The errors terms, $\varepsilon_{i,t}$, are assumed standard Gaussian.

2. Scenario F.2: The errors are assumed heteroscedastic and have some cross-sectional dependence. In particular, $\varepsilon_{i,t} = 0.9\varepsilon^1_{i,t} + \delta_t\varepsilon^2_{i,t}$, where $\delta_t = 1$ if t is odd and zero otherwise, and the vectors $\boldsymbol{\varepsilon}^1 = (\varepsilon^1_{1,t}, \varepsilon^1_{2,t}, \ldots, \varepsilon^1_{N,t})'$ and $\boldsymbol{\varepsilon}^2 = (\varepsilon^2_{1,t}, \varepsilon^2_{2,t}, \ldots, \varepsilon^2_{N,t})'$ follows a multivariate Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma} = (\sigma_{i,j})$ with $\sigma_{i,j} = 0.3^{|i-j|}$.

3. Scenario F.3: The errors have serial and cross-sectional dependence. In particular, $\varepsilon_{i,t} = 0.2\varepsilon_{i,t-1} + e_{i,j}$ where the vector $\boldsymbol{e}_t = (e_{1,t}, e_{2,t}, \ldots, e_{N,t})'$ follows a multivariate Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma} = (\sigma_{i,j})$ with $\sigma_{i,j} = 0.3^{|i-j|}$.

Tables 5 and 6 report the means of the ARI measure from 1000 replicates for these three scenarios when $N = 300$ and $T = 100$ and 200 and Tables 7 and 8 for $N = 600$ and $T = 100$ and 200, respectively. Additionally, the Tables 9–12 provides information on selected k as well as the percentage of times where $k^u = k_l$. In these scenarios, the selected number of lags, k, concentrates in the range 1–3 when $T = 100$ and in the range 0–2 when $T = 200$. The percentage of times where $k^u = k_l$ was very low, which produces a moderate-hight computational cost due the estimation of regressions (24). It should be notice that we are estimating, at least, 89700 regressions when $N = 300$ (359400 regressions when $N = 300$). The k selection step takes around 20.5 seconds when $N = 300$ (82.2 seconds when $N = 600$) for each replicate using a personal computer with an Intel(R) Core(TM) i7 CPU 920 2.67GHz processor.

The three univariate measures have a poor performance on dependency clustering, as expected, since all series have the same autocorrelation structures. Also, as expected, their behavior do not improve with the sample size. The DCN measure also have a poor performance on dependency clustering. Here, we should to realize that DCN procedure considers two time series that are negatively correlated away. This is the case, for instance, when the factor loadings, $\boldsymbol{\lambda}_{g_i,i}$, in model (26) have different sign than the factor loading, $\boldsymbol{\lambda}_{g_{i'},i'}$ with $g_i = g_{i'}$, that is, the ith and i'th series belong to the same group but DCN concludes that they are far away. As in the case of univariate measure, it should be noticed that DCN is not designed for cross-dependency clustering.

The proposed measure GCC (with single and average linkages) has an almost perfect classification when $T = 200$. The GCC-single, GCC-average and PAM-GCC outperform ABC. It is surprising that they work better than ABC, as scenarios F.1 – F.3 satisfy the required conditions for the ABC and the method use an iterative sophisticated estimation procedure. This result suggests that if we modify the clustering procedure built in the ABC procedure, that is, instead of clustering by a k-means algorithm applied to the estimated loadings we use the GCC measure the procedure will improve. To confirm this intuition, we perform an additional exercise where we run the full procedure proposed by Ando and Bai (2016) by now the initial clustering solution was obtained by the GCC-single, as proposed in this paper. We denote this procedure by ABC-GCC and the results are given in Tables 5–8. It is clear the improvement obtained by the new measure.

Table 5: Clustering performance evaluation (Adjusted Rand Index) at scenarios F.1 – F.3, $T = 100$ and $N_1 = N_2 = N_3 = 100$.

Method	Scenario F.1	Scenario F.2	Scenario F.3
QAC single	0.000	0.000	0.000
SAC single	0.001	0.001	0.001
PAC single	0.002	0.001	0.001
DCN single	0.000	0.000	0.000
TC single	0.962	0.946	0.969
GCC single	0.969	0.952	0.964
QAC complete	0.046	0.046	0.045
SAC complete	0.116	0.116	0.111
PAC complete	0.113	0.117	0.113
DCN complete	0.033	0.034	0.032
TC complete	0.362	0.359	0.351
GCC complete	0.312	0.309	0.306
QAC average	0.024	0.023	0.026
SAC average	0.102	0.095	0.096
PAC average	0.102	0.097	0.096
DCN average	0.020	0.020	0.020
TC average	0.992	0.988	**0.996**
GCC average	**0.993**	**0.992**	0.993
PAM - TC	0.837	0.821	0.819
PAM - GCC	0.912	0.908	0.912
ABC	0.876	0.877	0.868
ABC - GCC	0.991	0.990	0.990

Table 6: Clustering performance evaluation (Adjusted Rand Index) at scenarios F.1 – F.3, $T = 200$ and $N_1 = N_2 = N_3 = 100$.

Method	Scenario F.1	Scenario F.2	Scenario F.3
QAC single	0.000	0.000	0.000
SAC single	0.002	0.002	0.002
PAC single	0.002	0.002	0.002
DCN single	0.000	0.000	0.000
TC single	0.991	0.990	0.994
GCC single	0.993	0.990	0.991
QAC complete	0.043	0.043	0.043
SAC complete	0.098	0.094	0.094
PAC complete	0.098	0.096	0.094
DCN complete	0.035	0.037	0.034
TC complete	0.392	0.409	0.400
GCC complete	0.345	0.358	0.356
QAC average	0.023	0.021	0.021
SAC average	0.090	0.087	0.080
PAC average	0.089	0.084	0.079
DCN average	0.022	0.022	0.021
TC average	0.998	0.997	**1.000**
GCC average	0.997	0.998	0.998
PAM - TC	0.978	0.976	0.969
PAM - GCC	0.988	0.986	0.987
ABC	0.887	0.900	0.881
ABC - GCC	**0.999**	**0.999**	0.997

Table 7: Clustering performance evaluation (Adjusted Rand Index) at scenarios F.1 – F.3, $T = 100$ and $N_1 = N_2 = N_3 = 200$.

Method	Scenario F.1	Scenario F.2	Scenario F.3
QAC single	0.000	0.000	0.000
SAC single	0.000	0.001	0.000
PAC single	0.000	0.000	0.000
DCN single	0.000	0.000	0.000
TC single	0.915	0.901	0.937
GCC single	0.920	0.910	0.934
QAC complete	0.044	0.042	0.043
SAC complete	0.117	0.117	0.116
PAC complete	0.119	0.119	0.116
DCN complete	0.029	0.031	0.028
TC complete	0.360	0.360	0.368
GCC complete	0.302	0.313	0.316
QAC average	0.019	0.019	0.020
SAC average	0.097	0.095	0.089
PAC average	0.100	0.093	0.092
DCN average	0.018	0.018	0.017
TC average	0.983	0.974	0.992
GCC average	0.989	0.983	0.987
PAM - TC	0.802	0.785	0.780
PAM - GCC	0.912	0.907	0.908
ABC	0.880	0.886	0.884
ABC - GCC	**0.991**	**0.991**	**0.993**

Table 8: Clustering performance evaluation (Adjusted Rand Index) at scenarios F.1 – F.3, $T = 200$ and $N_1 = N_2 = N_3 = 200$.

Method	Scenario F.1	Scenario F.2	Scenario F.3
QAC single	0.000	0.000	0.000
SAC single	0.000	0.000	0.000
PAC single	0.000	0.000	0.000
DCN single	0.000	0.000	0.000
TC single	0.980	0.973	0.987
GCC single	0.982	0.974	0.980
QAC complete	0.047	0.046	0.045
SAC complete	0.106	0.107	0.103
PAC complete	0.107	0.104	0.102
DCN complete	0.032	0.034	0.031
TC complete	0.379	0.385	0.371
GCC complete	0.331	0.346	0.324
QAC average	0.022	0.019	0.020
SAC average	0.097	0.096	0.083
PAC average	0.096	0.096	0.083
DCN average	0.019	0.019	0.018
TC average	0.996	0.995	0.999
GCC average	**0.997**	**0.997**	0.997
PAM - TC	0.971	0.967	0.956
PAM - GCC	0.989	0.987	0.989
ABC	0.893	0.906	0.882
ABC - GCC	0.996	**0.997**	**1.000**

Table 9: Frequency of selected k at scenarios F.1 – F.3, $T = 100$ and $N_1 = N_2 = N_3 = 100$.

Selected k	Scenario F.1	Scenario F.2	Scenario F.3
0	4.70%	5.20%	3.10%
1	34.70%	31.40%	34.40%
2	38.40%	37.40%	37.20%
3	16.60%	20.40%	19.20%
4	4.70%	4.90%	5.50%
5	0.90%	0.70%	0.60%
$k^u = k_l$	4.40%	4.20%	4.40%

Table 10: Frequency of selected k at scenarios F.1 – F.3, $T = 200$ and $N_1 = N_2 = N_3 = 100$.

Selected k	Scenario F.1	Scenario F.2	Scenario F.3
0	16.50%	15.80%	6.70%
1	53.60%	55.80%	58.60%
2	26.40%	24.50%	26.70%
3	3.40%	3.30%	7.60%
4	0.10%	0.60%	0.40%
5	0.00%	0.00%	0.00%
$k^u = k_l$	3.20%	3.50%	3.00%

Table 11: Frequency of selected k at scenarios F.1 – F.3, $T = 100$ and $N_1 = N_2 = N_3 = 200$.

Selected k	Scenario F.1	Scenario F.2	Scenario F.3
0	1.90%	1.60%	0.50%
1	28.20%	25.10%	27.20%
2	35.50%	35.70%	34.80%
3	23.50%	25.50%	23.00%
4	9.10%	10.60%	12.20%
5	1.80%	1.50%	2.30%
$k^u = k_l$	3.60%	2.80%	3.50%

Table 12: Frequency of selected k at scenarios F.1 – F.3, $T = 200$ and $N_1 = N_2 = N_3 = 200$.

Selected k	Scenario F.1	Scenario F.2	Scenario F.3
0	9.90%	7.20%	2.20%
1	49.30%	50.00%	55.80%
2	32.80%	34.20%	28.80%
3	7.10%	7.50%	10.30%
4	0.90%	1.00%	2.80%
5	0.00%	0.10%	0.10%
$k^u = k_l$	3.70%	3.90%	3.20%

Note that both GCC and ABC improve their performance when T increases, and ABC also improve when N increases since the estimation method used in ABC benefits from a large number of series. GCC is slightly affected by the increase of N but this effect disappears when T increases. Essentially, GCC have comparable results with respect to ABC without using the information about the data generating model. The best results are obtained with ABC-GCC which combines a better starting point provided by the GCC and the use of a model-driven approach.

7 Clustering Electricity Prices

In this section, we consider a set of time series of hourly day-ahead prices for the New England electric market from January 2004 to December 2016. The data set is available at `www.iso-ne.com`. The New England region is divided in eight load zones: Connecticut (CT), Maine (ME), New Hampshire (NH), Rhode Island (RI), Vermont (VT), Northeastern Massachusetts and Boston (NEMA), Southeastern Massachusetts (SEMA) and Western/Central Massachusetts (WCMA). Each of the time series corresponds to the price of electricity in one of the eight regions at one of the 24 hours at a given weekday. For example, the first series corresponds to the price of electricity of the hour 1:00, on Thursday (since 01/01/2004 was Thursday), in the first region. Thus, we have 24x7x8=1344 time series of length 678 weeks. Notice that we have 365 (or 366) days each year for 13 years, making a total of 4749 days which are approximately 678 weeks. As an example, in Figure 3, we represent the series of prices corresponding to hour 12th of Thursdays for the eight load zones and the aggregated load zone. We observed a common pattern in the evolution of these series.

7.1 Analysis of the aggregated zone series

In this section, we will study the 24x7=168 series corresponding to the aggregated zone. The analysis of this small data set will provide us some insights with respect to the larger one with 1344 series corresponding to the eight load zones, that will analyzed next. First, we take a "regular" difference of the series:

$$X_{d,h}(w) = \log P_{d,h}(w) - \log P_{d,h}(w-1),$$

where $P_{d,h}(w)$ denotes the price at weekday d, hour h, and week w with $d = 1, 2, \ldots, 7$, $h = 1, 2, \ldots, 24$, and $w = 2, 3, \ldots, 678$. Notice that a "regular" difference in these series corresponds to a weekly seasonal difference in daily time series. It is well known that electricity price series have a strong weakly seasonality (see García-Martos and Conejo, 2013) and this transformation will produce stationary time series. Some authors have argued that the electricity prices series exhibit a long memory behaviour (see, for instance, Koopman et al, 2007) but, to simplify the example, we do not explore this possibility and assume that the series $X_{d,h}$ are stationary.

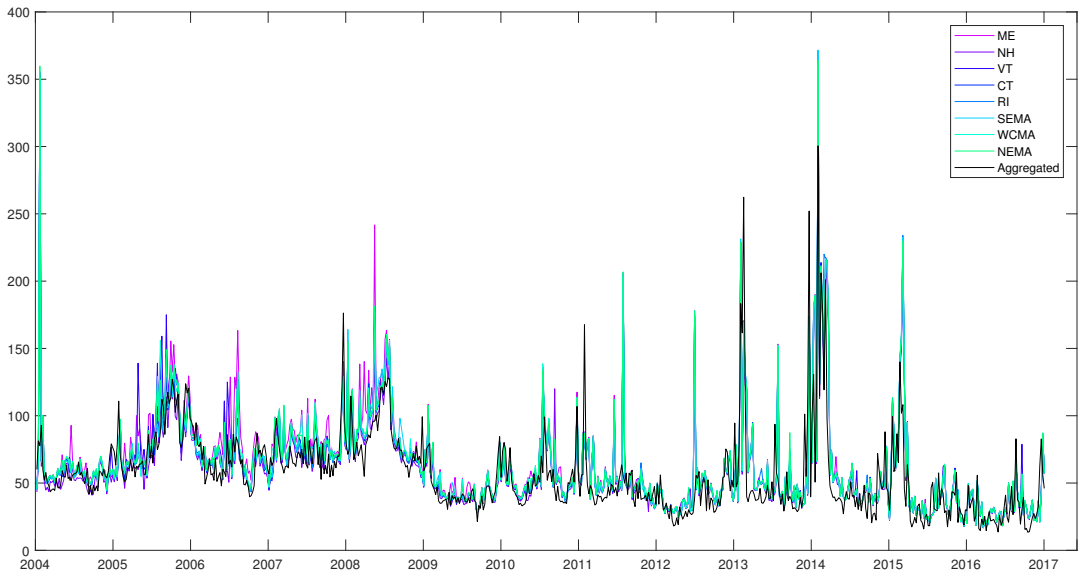

Figure 3: Prices at 12:00 on Thursdays for the eight load regions and the aggregated load zone of ISO New England market from January 2004 to December 2016.

A hierarchical clustering procedure (single linkage) using the GCC measure to the 168 series, $X_{d,h}$, gives the dendrogram presented in Figure 5. In this case, the selected number of lags was $k = 9$ ($k_l = 7$, $k^u = 9$ and the maximum considered lag was 35). This figure shows clear groups associated by the same weekday. This is to be expected, since the 24 hourly prices of a given day are simultaneously fixed in the daily market, producing a high cross-dependency. On the other hand, the univariate clustering and DCN procedures fails to detect this structure and make groups which are not related to the weekday. The adjusted Rand index for a seven clusters solution using QAC, SAC, PAC, DCN and GCC were 0.000, 0.002, 0.000, 0.225 and 0.819, respectively.

The number of clusters are selected by the Silhouette statistics (Rousseeuw, 1987) and the GAP procedure (Tibshirani et al, 2001) and both lead to 13 clusters (see Figure 4). It should be noted that the Silhouette statistic is more conclusive than the GAP statistic, since the latter increases again after 19 clusters. As mentioned in Tibshirani et al (2001), this could suggest 13 well-separated clusters and more less-separated ones. These clusters result from the fact that the procedure first divides by day of the week (weekday) and second each day is split into sleeping and awake hours: (i) sleeping hours, 01th-06th (or 01th-07th on weekend), and (ii) awake hours, 07th-24th (08th to 24th on weekend). It

should be noticed that a series in a cluster is not necessarily independent to the series in another cluster. For instance, the series corresponding to hour 01th is not independent of series of hour 24th but since they are in different clusters, we can infer that the dependence between the series of hour 01th with, at least, one series in the set of hours 02th-06th is higher than the dependence between the series of hours 01th and 24th.

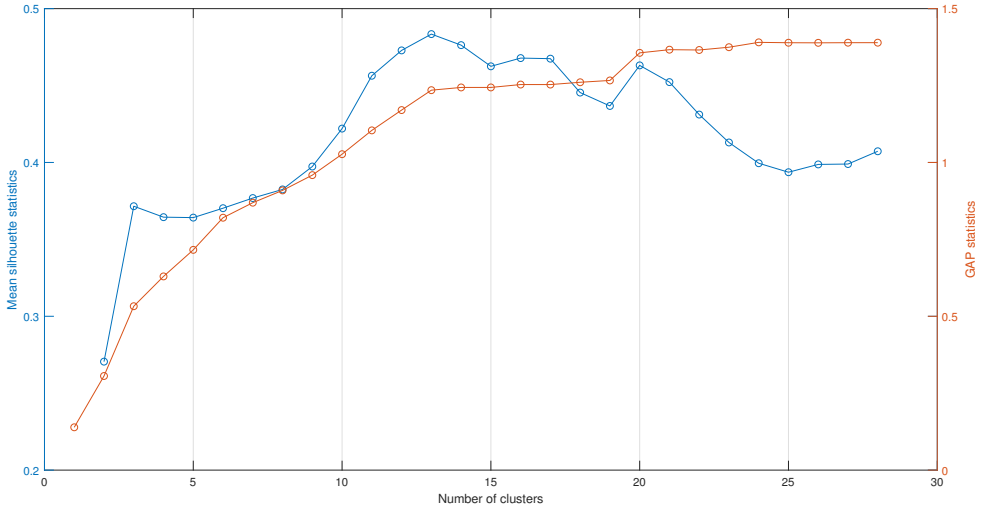

Figure 4: Silhouette and GAP statistics. Dataset of 168 series from aggregated load zone.

In order to gain insight on the obtained clusters, we will represent the set of time series as points in an ellipsoid graph where the nodes represent individual series and an edge between two nodes indicates strong cross-correlations as measured by a large value of the GCC measure (see Figure 6). That is, an edge between nodes i and j appears if $GCC(X_i, X_j)$ is bigger than a given threshold. The threshold used to obtain Figure 6 is based on the dendrogram in Figure 5. In particular, we select the cutoff threshold, 0.162, such that the series are divided into two groups. In Figure 6, we observed that the "sleeping hour" cluster exhibits an strong dependency, that is, all hours appear to be connected. Also, for the "awake hours" cluster we observed strong dependency among the series of hours from 10th to 22th but the dependencies decrease at the the hours at the frontiers of this cluster, that is the series corresponding to 07th–09th (08th–09th during weekend) and 23th–24th, that is, these series appear to have a chained dependence structure. Moreover, we observe that these dependencies change across the different days. This graph provides complementary information to the hierarchical structure of the dendrogram since it visualizes the dependencies between the series in the group.

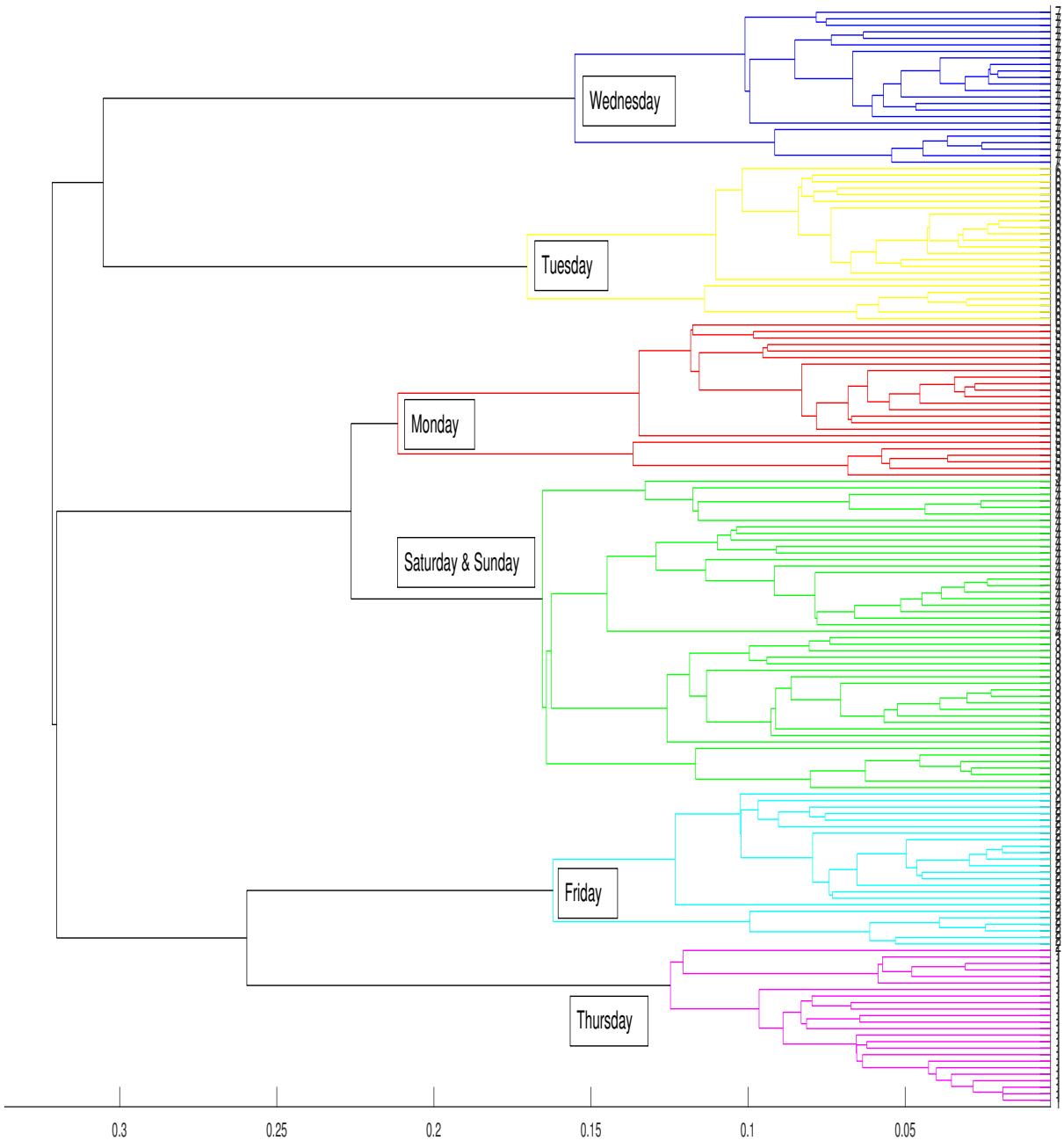

Figure 5: Dendrogram obtained with the regular differentiated series, $X_{d,h}$, of the aggregated zone.

Although the simulation experiments suggest that single linkage is preferable in presence of chained dependence structure, we also consider the complete and the average linkage. The adjusted Rand index for a seven clusters solution using single linkage versus complete and average linkage were 0.6495 and 0.8574, respectively. These values reveal a good level of concordance among the obtained clusters solutions.

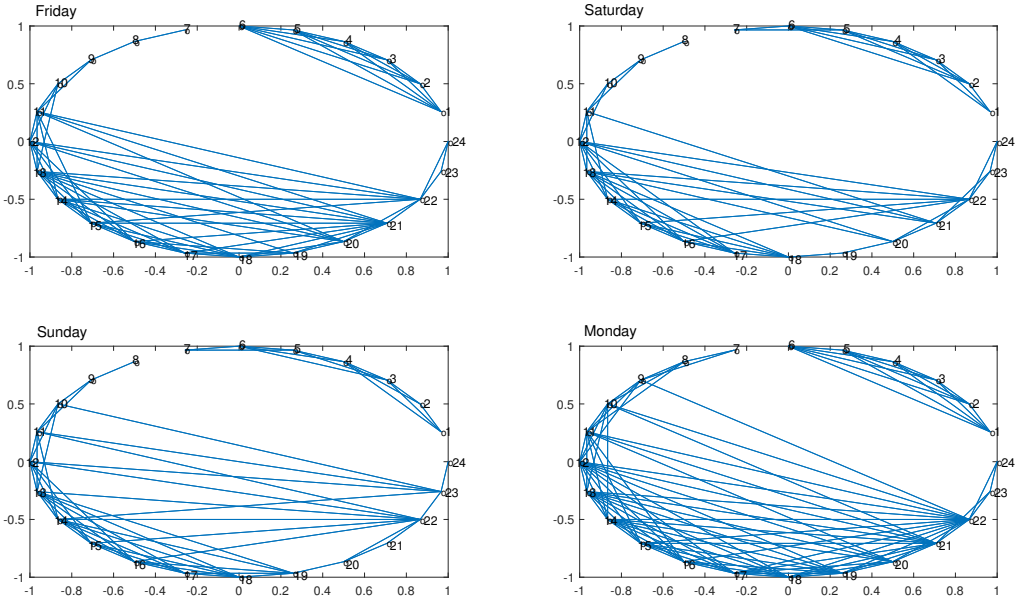

Figure 6: Undirected graphs for selected day's series of the aggregated zone.

7.2 Analysis of the eight load zones series

In this section, we will consider the 1344 time series corresponding to the eight load zones, that is, we have 168 series for each load zone. Figure 3 shows some examples of these series. As in the previous section, first, we take a regular difference of the series:

$$X_{d,h,z}(w) = \log P_{d,h,z}(w) - \log P_{d,h,z}(w-1),$$

where $P_{d,h,z}(w)$ denotes the price at weekday d, hour h, and week w at load zone z with $d = 1, 2, \ldots, 7$, $h = 1, 2, \ldots, 24$, $w = 2, 3, \ldots, 678$ and $z = 1, 2, \ldots, 8$. Secondly, we apply a hierarchical clustering procedure (single linkage) by using the GCC measure to the 1344 regular differentiated series, $X_{d,h,z}$, obtaining the dendrogram presented in Figure 9. We find a similar pattern of clusters than in the aggregated load zone.

In this data set, the Silhouette statistics and the GAP procedure provide different but plausible number of clusters, fourteen and seven, respectively. (see Figure 7). Again, GAP statistics have additional increases suggesting more than seven well-separated clusters. The seven groups corresponds to the five working days, the weekend and an additional division of the Monday's series. The fourteen groups in addition to the weekday takes into account two groups of hours in each day: (i) sleeping hours, 01th-06th (or 01th-07th on weekend), and (ii) the awake hours, 07th-24th (08th to 24th on weekend). There is a cluster conformed by a single series (Monday, 10am, Maine). Figure 8 shows the series of prices corresponding to hour 10th of Mondays for the eight load zones and it is clear that the different pattern in Maine's series is due to the presence of a few very large outliers.

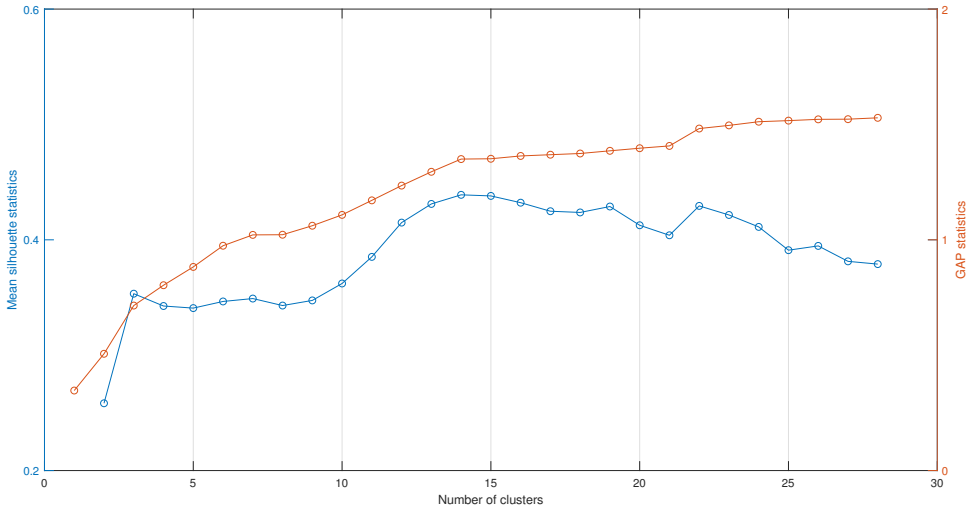

Figure 7: Silhouette and GAP statistics. Dataset of 1344 series from the eight load regions.

Additionally, we have checked that the clusters found by the hierarchical clustering procedure are similar to the ones obtained by a k-means algorithm applied to the main principal coordinates using multidimensional scaling on matrix (25). 280 principal coordinates are selected that account for more than 90% of the variability. The adjusted Rand index for a seven and fourteen clusters solutions of hierarchical clustering (single linkage) and k-means procedures were 0.734 and 0.795, respectively. Figure 10 shows the biplot of the series on the two main principal coordinates (they account for 51% of the variability). Again, the groups associated to the same weekday are clearly identifiable.

As a conclusion, we observed that both clustering procedures using GCC measure reveal some interesting features of this large set of time series.

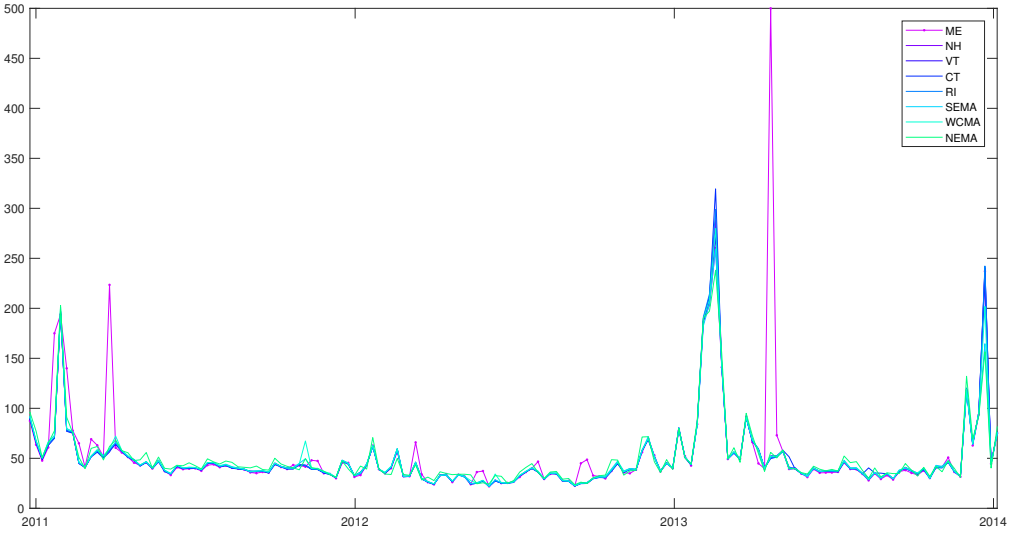

Figure 8: Prices at 10:00 on Monday for the eight load regions ISO New England market from January 2011 to December 2014.

8 Conclusions

We have presented a novel procedure for clustering time series that takes into account their cross dependency. The procedure is based on pairwise measures involving the determinant of the cross correlation matrices from lag zero to lag k. A Monte Carlo study has shown the good performance of the cluster methods with respect to some alternatives.

The proposed procedure can be used both in exploratory analysis of a large set of time series that will help the modelling and, also, as a tool for improving the forecast of large sets of time series as studied with model based cluster by Wang et al (2013). It has been shown that the proposed cluster methods can be very useful in building dynamic factor models with cluster structure, as studied by Ando and Bai (2016, 2017).

The results in this article can be extended in several directions. First, as it is well known and it has been shown in the electricity prices, cluster procedures can be sensitive to outliers and the proposed procedure can be improved by: (1) Using robust estimation of the cross-correlation coefficients or; (2) Applying an outlier cleaning method to the set of time series before the clustering is carried out. Second, for very large data sets other clustering algorithms can be applied, as those based on projections. Third, the method can be extended for some types of non linearity in time series. In the case of conditionally heteroscedastic time series the extension seems to be straightforward by using the cross correlation

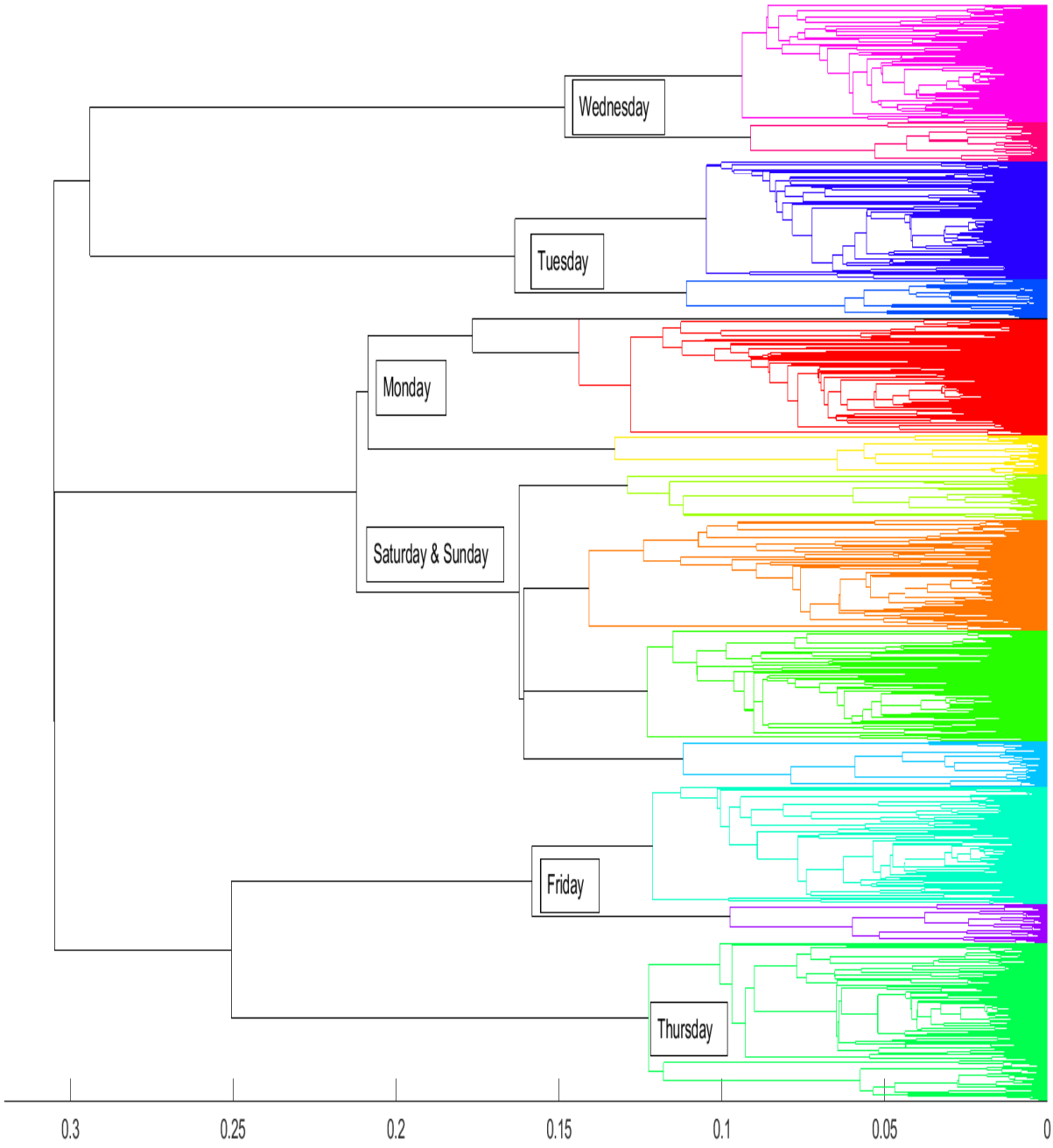

Figure 9: Dendrogram obtained with the regular differentiated series, $X_{d,h,z}$, of the eight load zones.

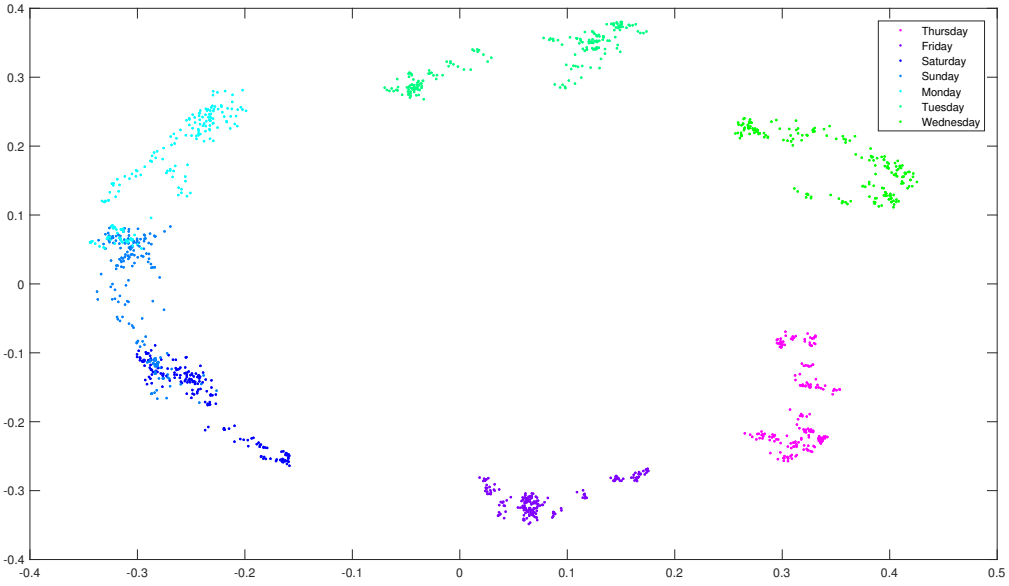

Figure 10: Biplot of the series on the two main principal coordinates.

between squared residuals, but other more general types of nonlinearity could be considered. These problems will be the subject of further research.

9 Acknowledgements

This research has been supported by Grant ECO2015-66593-P of MINECO/FEDER/UE. We thanks to professor Tomohiro Ando for making available their code and kindly answer some questions regarding the implementation.

References

[1] Aghabozorgi, S. and Wah, T. Y. Clustering of large time series data sets. *Intell. Data Anal.*, **18**, 793–817 (2014).

[2] Aghabozorgi, S., Shirkhorshidi, A. S. and Wah, T.Y. Time-series clustering – A decade review. *Inform. Syst.*, **53**, 16–38 (2015).

[3] Alonso, A. M., Berrendero, J. R., Hernández, A. and Justel, A. Time series clustering based on forecast densities. *Comput. Stat. Data. An.*, **51**, 762–766 (2006).

[4] Anderson, T. W. *An introduction to multivariate statistical analysis*, New York, John Wiley, 2nd edition (1984).

[5] Ando, T. and Bai J. Panel data models with grouped factor structure under unknown group membership. *J. Appl. Econom.*, **31**, 163–191 (2016).

[6] Ando, T. and Bai J. Clustering huge number of financial time series: A panel data approach with high-dimensional predictor and factor structures. *J. Am. Statist. Ass.*, **112**, 1182–1198 (2017).

[7] Caiado, J., Crato, N. and Peña, D. A periodogram-based metric for time series classification. *Comput. Stat. Data. An.*, **50**, 2668–2684 (2006).

[8] Caiado, J., Maharaj, E. A., and D'Urso, P. Time series clustering. *Handbook of cluster analysis*. Chapman and Hall/CRC (2015).

[9] Corduas, M. and Piccolo, D. Time series clustering and classification by the autoregressive metric. *Comput. Stat. Data. An.*, **52**, 1860-1872 (2008).

[10] Davidson, J. *Stochastic Limit Theory. An Introduction for Econometricians*, London, Oxford University Press (1994).

[11] Douzal-Chouakria, A. and Nagabhushan, P.N. Adaptive dis- similarity index for measuring time series proximity. *Adv. Data Anal. Classif.*, **1**, 5–21, (2007).

[12] D'Urso, P. and Maharaj, E.A. Autocorrelation-based Fuzzy Clustering of Time Series, *Fuzzy Set. Syst.*, **160**, 3565–3589, (2009).

[13] D'Urso, P., Maharaj, E.A. and Alonso, A.M. Fuzzy clustering of time series using extremes, *Fuzzy Set. Syst.*, **318**, 56–79, (2017).

[14] Fruhwirth-Schnatter, S. and Kaufmann, S. Model-Based Clustering of Multiple Time Series. *J. Bus. Econ. Statist.*, **26**, 78–89 (2008).

[15] García-Martos, C. and Conejo, A.J. Price forecasting techniques in power system,J. Webster (ed.) Wiley *Encyclopedia of Electrical and Electronics Engineering*. John Wiley & Sons (2013).

[16] Golay, X., Kollias, S., Stoll, G., Meier, D., Valavanis, A. and Boesiger, P. A new correlation-based fuzzy logic clustering algorithm for fmri. *Magn. Reson. Med.*, **40**, 249–260 (2005).

[17] Granger, C. W. and Morris, M. J. Time series modelling and interpretation. *J. R. Statist. Soc. A*, **139**, 246–257 (1976).

[18] Hallin, M. and Lippi, M. Factor models in high-dimensional time series—A time-domain approach. *Stoch. Proc. Appl.*, **123**, 2678–2695 (2013).

[19] Hamilton, J. D. *Time Series Analysis*, New Jersey, Princeton University Press (1994).

[20] Hannan, E. J. *Multiple Time Series*, New York, John Wiley& Sons (1970).

[21] Hennig, C. Clustering strategy and method selection. In Hennig, C., M. Meila, F. Murtagh, and R. Rocci (Eds.). *Handbook of Cluster Analysis*. Chapman and Hall/CRC, 703–730 (2015).

[22] Hubert, L. and Arabie, P. Comparing partitions. *J. Classification*, **2**, 193–218 (1985).

[23] Kakizawa, Y., Shumway, R. H. and Taniguchi, M. Discrimination and Clustering for Multivariate Time Series. *J. Am. Statist. Ass.*, **93**, 328–340 (1998).

[24] Knapp, C. and Carter, G. The generalized correlation method for estimation of time delay. *IEEE T. Acoust. Speech*, **24**, 320–327 (1976).

[25] Koopman, S. J., Ooms, M. and Carnero, M. A. Periodic Seasonal Reg-ARFIMA-GARCH Models for Daily Electricity Spot Prices. *J. Am. Statist. Ass.*, **102**, 16–27 (2007).

[26] Kullback, S. *Information Theory and Statistics*. Dover (1968).

[27] Lafuente-Rego, B. and Vilar, J. A. Clustering of time series using quantile autocovariances. *Adv. Data Anal. Classi.*, **10**, 391–415 (2015).

[28] Lam, C. and Yao, Q. Factor modeling for high-dimensional time series: inference for the number of factors. *Ann. Statist.*, **40**, 694–726 (2012).

[29] Larsen, B. and Aone, C. Fast and effective text mining using linear time document clustering, in: *Proceedings of the Conference on Knowledge Discovery and Data Mining*, 16–22 (1999).

[30] Liao, T. W. Clustering of time series data-a survey. *Pattern Recog.*, **38**, 1857–1874 (2005).

[31] Lütkepohl, H. *Handbook of matrices*, John Wiley&Sons (1996).

[32] Maharaj, E. A. Comparison of non-stationary time series in the frequency domain. *Comput. Stat. Data. An.*, **40**, 131–141 (2002).

[33] Maharaj, E. A. and D'Urso, P. Fuzzy Clustering of Time Series in the Frequency Domain, *Inform. Sciences*, **181**, 1187–1211, (2011).

[34] Mahdi, E. and McLeod, I. A. Improved multivariate portmanteau test. *J. Time Ser. Anal.*, **33**, 211–222 (2012).

[35] Meilă, M. Comparing clusterings – An information based distance. *J. Multivariate Anal.*, **98** 873–895 (2007).

[36] Montero, P. and Vilar, J. TSclust: An R package for time series clustering. *J. Stat. Softw.*, **62**, 1–43, (2014).

[37] Pamminger, C. and Fruhwirth-Schnatter, S. Model-based Clustering of Categorical Time Series. *Bayesian Anal.*, **2**, 345–368 (2010).

[38] Peña, D. and Box, G. E. P. Identifying a simplifying structure in time series. *J. Am. Statist. Ass.*, **82**, 836-843 (1987).

[39] Peña, D. and Rodríguez, J. A powerful portmanteau test of lack of test for time series. *J. Am. Statist. Ass.*, **97**, 601–610 (2002).

[40] Peña, D., and Rodríguez, J. Descriptive measures of multivariate scatter and linear dependence. *J. Multivariate Anal.*, **85**, 361–374 (2003).

[41] Pértega, S. and Vilar, J. A. Comparing Several Parametric and Nonparametric Approaches to Time Series Clustering: A Simulation Study. *J. Classif.*, **27**, 333–362 (2010).

[42] Piccolo, D. A distance measure for classifying ARMA models. *J. Time Ser. Anal.*, **2**, 153–163 (1990).

[43] Robbins, M. W. and Fisher, T. J. Cross-Correlation Matrices for Tests of Independence and Causality Between Two Multivariate Time Series. *J. Bus. Econ. Statist.*, **33**, 459–473 (2015).

[44] Rousseeuw, P. J. Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Comput. Appl. Math.*, **20**, 53–65 (1987).

[45] Sadahiro, Y. and Kobayashi, T. Exploratory analysis of time series data: Detection of partial similarities, clustering, and visualization. *Comput. Environ. Urban*, **45**, 24–33 (2014).

[46] Scotto, M. G., Barbosa, S. M. and Alonso, A. M. Extreme value and cluster analysis of European daily temperature series *J Applied Stat*, **38**, 2793–2804, (2011).

[47] Tibshirani, R. Regression Shrinkage and Selection via the lasso. *J. R. Statist. Soc. B*, **58**, 267–288 (1996).

[48] Tibshirani, R., Walther, G. and Hastie, T. Estimating the number of clusters in a data set via the gap statistic. *J. R. Statist. Soc. B*, **63**, 411–423 (2001).

[49] Vilar-Fernández, J. A., Alonso, A. M. and Vilar-Fernández, J. M. Nonlinear time series clustering based on nonparametric forecast densities. *Comput. Stat. Data. An.*, **54**, 2850–2865 (2010).

[50] Vilar, J. A., Lafuente-Rego, B. and D'Urso, P. Quantile autocovariances: a powerful tool for hard and soft partitional clustering of time series, *Fuzzy Set. Syst.*, **340**, 38–72, (2018).

[51] Wang, Y., Tsay, R. S., Ledolter, J., and Shrestha, K. M. Forecasting Simultaneously High-Dimensional Time Series: A Robust Model-Based Clustering Approach. *J. Forecasting*, **32**, 673–684 (2013).

[52] Xiong, Y. and Yeung, D. Time series clustering with ARMA mixtures. *Pattern Recog.*, **37**, 1675–1689 (2004).

[53] Zhang, X., Liu, J., Du, Y. and Lv, T. A novel clustering method on time series data. *Expert Syst. Appl.*, **38**, 11891–11900 (2011).

[54] Zhang, T. Clustering High-Dimensional Time Series Based on Parallelism. *J. Am. Statist. Ass.*, **108**, 577–588 (2013).