Electronic Journal of Statistics
Vol. 7 (2013) 1019–1046
ISSN: 1935-7524
DOI: 10.1214/13-EJS800

Time series clustering based on nonparametric multidimensional forecast densities

José A. Vilar[*,†] **and Juan M. Vilar**[†]

University of A Coruña
Department of Mathematics
Campus de Elviña, s/n, 15009 A Coruña, Spain
e-mail: jose.vilarf@udc.es; juan.vilar@udc.es

Abstract: A new time series clustering method based on comparing forecast densities for a sequence of $k > 1$ consecutive horizons is proposed. The unknown k-dimensional forecast densities can be non-parametrically approximated by using bootstrap procedures that mimic the generating processes without parametric restrictions. However, the difficulty of constructing accurate kernel estimators of multivariate densities is well known. To circumvent the high dimensionality problem, the bootstrap prediction vectors are projected onto a lower-dimensional space using principal components analysis, and then the densities are estimated in this new space. Proper distances between pairs of estimated densities are computed and used to generate an initial dissimilarity matrix, and hence a standard hierarchical clustering is performed. The clustering procedure is examined via simulation and is applied to a real dataset involving electricity prices series.

AMS 2000 subject classifications: Primary 62H30, 62M20; secondary 62G07, 62G09.
Keywords and phrases: Time series clustering, multidimensional forecast density, bootstrap, kernel estimation, principal components analysis.

Received December 2012.

Contents

[*] Corresponding author.

[†] Research supported by Grants 10DPI105003PR and CN2012/130 from Xunta de Galicia, and by Grant number MTM2011-22392 from Ministerio de Ciencia e Innovación (Spain).

1. Introduction

Clustering is an unsupervised learning process aimed to find similarities and differences among data objects to classify them into a small number of homogeneous groups. When the objects are time series data, this classification might be useful to detect a few representative patterns, quantify the affinity through the time, forecast future performances, find out unknown temporal patterns, etc. In particular, the problem of grouping together similar time series arises in a broad range of fields such as economics, finance, medicine, bioinformatics, ecology, geology, environmental studies, engineering, and many others. Some illustrative examples reported in the literature are: comparison of seismological data as the case of distinguishing between earthquake and nuclear explosions waveforms [19], clustering of industrialized countries according to historical data of CO_2 emissions [1], detection of similar immune response behaviors of CD4+ cell number progression over patients affected by immune deficiency virus (HIV) [7], clustering of banks on the basis of their weekly share price series [37], clustering of industrial production indices [34], the automatic identification of groups of rail switching operations by analyzing time series of electrical power consumption acquired during these operations [31], and many others. Note that time series data present specific features that make difficult the clustering task. First, time series data are dynamic in their nature, with an underlying autocorrelation structure, and hence the analysis of similarities between series should regard their evolution in time. In addition, time series database are usually formed by large amounts of records and most of the standard clustering algorithms do not work efficiently on high-dimensional data. Previous arguments motivate that the number of contributions on time series clustering has increased substantially in recent years, becoming a very active research area nowadays. A detailed and extensive review on time series clustering is given by Liao [24], who introduces the basics of this topic and provides a set of interesting references and some specific application areas along with the sources of data used.

One key issue in time series clustering is the choice of a suitable dissimilarity measure between two time series. Most conventional metrics used in cluster analysis are inherently static because they assess the closeness of the values observed in specific instants of time, ignoring the interdependence relationship between values. In fact, the concept of similarity between time series is not simple and it can be established in different ways. Corduas and Piccolo [9] (see Introduction and references therein) provide a valuable overview on the different approaches considered in the literature to construct dissimilarity measures between time series. One way is to directly compare observations or specific features extracted from raw data (see [23, 32, 11, 5, 7], among others). An alternative approach is to assess the discrepancy between the underlying generating processes (some references following this approach are [29, 25, 26, 19, 35], among many others). Thus, there exist a broad range of metrics to compare time series, but the question is: “which dissimilarity measure is the best?”. Although some works have compared the performance of some of these metrics via simulation studies [20, 3, 5, 28], it is clear that the choice of a suitable metric heavily relies on the nature of the clustering, i.e. on determining what the purpose of the grouping is.

In this work, the notion of dissimilarity is governed by the performance of future forecasts. More precisely, two time series are similar if their forecasts for a specific sequence of future times are close. This similarity concept could produce results very different to the ones coming from a cluster procedure based on the generating models or on the last observed values. For instance, two time series coming from the same generating process can lead to different forecasts at pre-specified time points, and hence they might not be clustered together. However, there are many practical situations where the real interest of the clustering relies directly on the properties of the predictions, as in the case of any sustainable development problem or in situations where the concern is to reach target values on pre-specified future time periods. Alonso et al. [1] consider this similarity notion and assume that there is just one future time point of interest, say $T+h$, where T denotes the length of the observed series and h is the horizon of interest. The dissimilarity measure between two time series is defined by the L^2 distance between their full forecast densities at the pre-specified horizon $T+h$. They argue that comparing the forecast densities allows us to take into account the variability of the predictions, which is completely ignored when the comparison is based on the point forecasts. The forecast densities are approximated using kernel-type density estimators based on a sufficiently large set of sieve-bootstrap predictions, which requires to assume that the series admit an AR(∞) representation. To overcome this limitation, Vilar et al. [34] extend their results to cover the case of nonparametric models of arbitrary autoregressions. In this new scenario, the sieve bootstrap is not valid, and hence the forecast densities are approximated using a bootstrap procedure that mimics the underlying generating processes without assuming any parametric model for the true autoregressive structure of the series. Furthermore, it is also shown that the L^1 distance presents better properties than the L^2 distance in this clustering setup.

Works by Alonso et al. [1] and Vilar et al. [34] focused on the case of only one forecast horizon, but in practice, the horizon of interest to perform clustering is frequently a period of $k > 1$ future times. For instance, the classification of European countries based on the future behavior of some economic indicators (inflation, real interest rate, trade balance, domestic credit growth, public debt, ...) should be useful to gain knowledge on the evolution of the current financial crisis in the European Union. However, a realistic classification should be based on a reasonable forecast period of two or more years. Analogously, classifying some stock market companies according to the predictions of their daily stock prices should be also of great interest for investors, as long as the predictions involve a period of several days or weeks and not just one future day. Grouping countries with similar forecasts of mortality rates for a range of years ahead, clustering regions in accordance with predictions of monthly temperature for the next annual cycle (i.e. $k = 12$), and many others real examples justify the interest of time series clustering based on a set of $k > 1$ predictions. Thus, the main goal in this work is to extend the methodology proposed by Vilar et al. [34] to the case of considering forecasts on a set of $k > 1$ consecutive future times, i.e. the similarity notion is now governed by the proximity between k-variate forecast densities. The resampling algorithms used in [34] are also valid to obtain bootstrap replications of sequences of k predictions, simply lengthening the bootstrap prediction-paths (see Section 2.1). Here, the main drawback is to construct adequate kernel estimators of the multivariate forecast densities. There exist well-known problems related to the multivariate kernel density estimation: high computational effort, specification of several bandwidth parameters, and mainly that it is complex to obtain accurate estimators in more than three or four dimensions due to the so-called *curse of dimensionality*. To circumvent the high dimensionality problem, we propose to transform the prediction vectors into a new set of outputs confined in a low-dimensional space, and then to compare the estimated densities in this new space.

The rest of the paper is organized as follows. Section 2 presents the steps of our clustering method and includes three subsections where basic aspects of the procedure are discussed in detail. Specifically, the approaches to obtain samples of bootstrap predictions are described in Section 2.1, the problem of reducing the dimensionality of the forecasts is addressed in Section 2.2, and the metrics to construct a pairwise dissimilarity matrix are introduced in Section 2.3. Section 3 reports the results from a complete simulation study designed to evaluate the performance of our clustering procedure. In Section 4, our clustering methodology is applied to real time series concerning electricity price in the Spanish electricity market. The dataset is formed by a collection of 24 series such that the i-th series contains the historical daily prices at hour i, for $i = 1, \ldots, 24$, and the purpose is to identify groups of hours with a similar behaviour of their one-week-ahead predictions. Some concluding remarks are stated in Section 5.

2. Description of the clustering procedure

Consider a set of time series $S = \left\{\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \ldots, \boldsymbol{X}^{(s)}\right\}$, where each element $\boldsymbol{X}^{(i)} = \left(X_1^{(i)}, \ldots, X_T^{(i)}\right)$ is a partial realization from a real valued stationary process $\{X_t^{(i)},\ t \in \mathbb{Z}\}$ that admits a general autoregressive representation of the form

$$X_t^{(i)} = m_i(X_{t-j_1}^{(i)}, X_{t-j_2}^{(i)}, \ldots, X_{t-j_d}^{(i)}) + \varepsilon_t^{(i)}, \tag{2.1}$$

where $(X_{t-j_1}^{(i)}, X_{t-j_2}^{(i)}, \ldots, X_{t-j_d}^{(i)})$, $j_1 < j_2 < \ldots < j_d$, is a d-dimensional vector of known lagged variables and $\{\varepsilon_t^{(i)}\}$ is a sequence of i.i.d. random variables. The unknown autoregressive functions $m_i(\cdot)$ are assumed to be smooth functions but they are not restricted to any pre-specified parametric model. Hence, both linear and nonlinear autoregressive processes might be included in S.

We are interested in performing cluster analysis on S in such a way that series showing a similar behaviour for their first k predictions, $k > 1$, are grouped together, i.e. our clustering must be governed by the performance of the predictors $\boldsymbol{X}_k^{(i)} = \left(X_{T+1}^{(i)}, \ldots, X_{T+k}^{(i)}\right)$. In practice, the number k of point predictions is known and previously fixed by practitioners according to the nature and the goals of the analyzed problem.

In this context, we adopt the dissimilarity concept considered in [1] and [34], namely the distance between two series $\boldsymbol{X}^{(i)}$ and $\boldsymbol{X}^{(j)}$ is measured in terms of the discrepancy between the bootstrap forecast densities of their predictors $\boldsymbol{X}_k^{(i)}$ and $\boldsymbol{X}_k^{(j)}$. However, this discrepancy is here evaluated in a lower-dimensional space where the bootstrap replicates of the predictors are previously projected. Specifically, we propose a clustering procedure involving the five steps indicated below.

`Step 1.` Generate a large set of bootstrap predictions for each series.
`Step 2.` Determine a low-dimensional space where the bootstrap predictions are projected.
`Step 3.` Obtain a multivariate kernel density estimate from each set of projected bootstrap predictions.
`Step 4.` Construct a dissimilarity matrix based on pairwise distances between the density estimates obtained in `Step 3`.
`Step 5.` Apply a standard hierarchical clustering method to the dissimilarity matrix.

As the forecast densities are unknown in practice, a large number of bootstrap predictors is obtained in `Step 1` to estimate these densities. Under more restrictive parametric assumptions, e.g. that the generating processes are AR(p) with Gaussian errors, the conditional density for the predictions h-steps-ahead, $X_{T+h}^{(i)}\big|_{X_1^{(i)},\dots,X_T^{(i)}}$ is asymptotically normal and can be directly estimated without using bootstrap. However, we seek a more general procedure that can be applied to a broad range of models. In particular it is desirable to overcome the normality and linearity restrictions, and bootstrap is a very useful device in this scenario. Although different resampling mechanisms can be used, we have selected three methods that allow us to relax gradually these restrictions and that are based on the common idea of mimicking the underlying dependence structure by resampling residuals (see Section 2.1 for details). Once the bootstrap samples are obtained, we could proceed as in [34], that is approximating the unknown forecast densities by means of kernel density estimators and computing the distances between each pair of estimators. However, as mentioned in the previous section, it is difficult to obtain accurate density estimators in more than three or four dimensions and hence this approach is inadequate for moderate or large values of k. For this reason, `Step 2` of the algorithm consists in projecting the bootstrap prediction vectors onto a lower-dimensional space. In particular, an approach based in principal components analysis (PCA) is considered to construct this new space of dimension as small as possible (see Section 2.2). In `Step 3`, the coordinates of the transformed bootstrap prediction vectors, i.e. the principal components scores, are used to obtain kernel estimators of the prediction densities in the new low-dimensional space. `Step 4` allows us to construct a pairwise dissimilarity matrix by computing the distances between each pair of estimators. Several metrics are proposed in Section 2.3 to assess these distances. Taking as starting point the dissimilarity matrix obtained in `Step 4`, a conventional hierarchical clustering method is finally performed in `Step 5` to obtain the required cluster solution.

The following subsections deal more in detail with the key points of the clustering algorithm: the resampling procedure required in `Step 1`, the approach to solving the dimensionality reduction problem considered in `Step 2`, and the metric used to assess the distance between two multivariate density estimators in `Step 4`.

2.1. Obtaining samples of bootstrap predictions

The first step of our algorithm consists in obtaining a large number of bootstrap replicates of the predictors $\boldsymbol{X}_k^{(i)} = \left(X_{T+1}^{(i)}, \ldots, X_{T+k}^{(i)}\right)$, $i = 1, \ldots, s$, in order to gain knowledge on their distributions. There are several approaches to perform bootstrap for prediction in a time series setup. If a particular structure is explicitly stated for the dependence (for instance, an ARIMA model), then the general idea is to construct a tailor designed resampling mechanism that takes into account the explicit way in which some observation depends on its past. The aim is that the bootstrap sample exhibits approximately the same pattern of dependence. Following this criterion and according to (2.1), we have considered three bootstrap methods based on mimicking an autoregressive structure.

Consider a partial realization $(X_1, \ldots, X_T)$ from a stationary process $\{X_t\}_{t\in\mathbb{Z}}$ satisfying (2.1). The first resampling scheme, so-called *autoregression bootstrap*, proceeds as follows.

`AB.1` Estimate the unknown autoregressive function $m(\cdot)$ from the original sequence $(X_1, \ldots, X_T)$ by using a truncated Nadaraya-Watson smoother with bandwidth g_1, say $\hat{m}_{g_1}(\cdot)$.

`AB.2` Compute nonparametric residuals, $\hat{\varepsilon}_t = X_t - \hat{m}_{g_1}(\boldsymbol{X}_{t-1})$, for $t = d + 1, \ldots, T$.

`AB.3` Construct the kernel density estimate with bandwidth h of the centered residuals, $\hat{f}_{\tilde{\varepsilon},h}$, where $\tilde{\varepsilon}_t = \hat{\varepsilon}_t - \hat{\varepsilon}_.$, with $\hat{\varepsilon}_.$ being the mean of the $\hat{\varepsilon}_t$.

`AB.4` Draw a sufficiently large bootstrap-resample ε_t^* of i.i.d. observations from $\hat{f}_{\tilde{\varepsilon},h}$.

`AB.5` Using the estimator $\hat{m}_{g_1}$ introduced in `AB.2`, define the bootstrap series X_t^*, $t = 1, \ldots, T$, by the recursion

$$X_t^* = \hat{m}_{g_1}(\boldsymbol{X}_{t-1}^*) + \varepsilon_t^*.$$

`AB.6` Estimate the bootstrap autoregressive function, m^*, using again a truncated smoother $\hat{m}_{g_2}(\cdot)$, with bandwidth g_2, based on the bootstrap series $(X_1^*, \ldots, X_T^*)$.

`AB.7` Compute bootstrap prediction-paths of length k by setting

$$X_t^* = \hat{m}_{g_2}^*(\boldsymbol{X}_{t-1}^*) + \varepsilon_t^*,$$

for $t = T + 1, T + 2, \ldots, T + k$, and $X_t^* = X_t$, for $t \leq T$.

`AB.8` Repeat steps `AB.1`-`AB.7` a large number (B) of times to obtain bootstrap replications of the predictors $\boldsymbol{X}_k = (X_{T+1}, \ldots, X_{T+k})$.

A more detailed description of this bootstrap procedure can be seen in [10], where the consistency of the method is established and the choice of the pilot estimate $\hat{m}_{g_1}(\cdot)$ and the auxiliary bandwidths g_1 and g_2 is also discussed. In our experiments, the cross-validation bandwidth selector introduced by Hart [15] was used to obtain g_1 because this automatic selector is specifically designed to deal with dependent data. Following the suggestion by Franke et al. [10], g_2 was taken to be larger than g_1, such as $g_2 = 1.5g_1$ or $g_2 = 2g_1$.

Unlike other residual-based resampling mechanisms, such as the called *sieve bootstrap* (see e.g. [4, 2]), the autoregression bootstrap does not assume a parametric structure for the underlying autoregressive function, which results in a great versatility that enables it to be applied to a broad range of models. The sieve bootstrap, in contrast, assumes a linear structure for the underlying autoregressive process, $m(\boldsymbol{x}) = \boldsymbol{\phi}^t\boldsymbol{x}$, and hence the residuals are generated from an estimator of a stationary $AR(d)$ process. In this way, the sieve bootstrap can be adversely affected by departures of the linearity assumption. To assess this effect, the sieve bootstrap was also considered in our simulation study in Section 3. Briefly, the sieve bootstrap algorithm can be outlined as follows.

SB.1 Obtain the least squares estimates of the autoregressive coefficients $\hat{\boldsymbol{\phi}} = (\hat{\phi}_1, \ldots, \hat{\phi}_d)$.

SB.2 Compute the least square residuals given by $\hat{\varepsilon}_t = \sum_{j=0}^{d} \hat{\phi}_j L^j X_t$, where L denotes the lag operator, i.e. $L^j X_t = X_{t-j}$.

SB.3 Construct the kernel density estimate with bandwidth h of the centered residuals, $\hat{f}_{\tilde{\varepsilon},h}$, where $\tilde{\varepsilon}_t = \hat{\varepsilon}_t - \hat{\varepsilon}_{\cdot}$, with $\hat{\varepsilon}_{\cdot}$ being the mean of the $\hat{\varepsilon}_t$.

SB.4 Draw a sufficiently large bootstrap-resample ε_t^* of i.i.d. observations from $\hat{f}_{\tilde{\varepsilon},h}$.

SB.5 Define the bootstrap series X_t^*, $t = 1, \ldots, T$, by the recursion

$$\sum_{j=0}^{d} \hat{\phi}_j L^j X_t^* = \varepsilon_t^*.$$

SB.6 Using the bootstrap resample obtained in SB.5, compute the least squares estimates of the bootstrap autoregressive coefficients $\hat{\boldsymbol{\phi}}^* = (\hat{\phi}_1^*, \ldots, \hat{\phi}_d^*)$.

SB.7 Compute the bootstrap prediction-paths of length k by setting

$$\sum_{j=0}^{d} \hat{\phi}_j^{\star} L^j X_t^* = \varepsilon_t^*,$$

for $t = T+1, T+2, \ldots, T+k$, and $X_t^* = X_t$, for $t \leq T$.

SB.8 Repeat steps SB.1-SB.7 a large number (B) of times to obtain bootstrap replications of the predictors $\boldsymbol{X}_k = (X_{T+1}, \ldots, X_{T+k})$.

The third resampling plan considered in the present work is a *conditional bootstrap* method proposed in Cao et al. [6]. The idea consists in modifying the autoregression bootstrap algorithm as follows: (i) steps AB.5 and AB.6 are omitted, and (ii) the bootstrap prediction-paths computed in AB.7 are generated using $\hat{m}_{g_1}$ instead of $\hat{m}_{g_2}$. Equivalently, the resamples of the future values are obtained conditionally to the original series. The conditional bootstrap is also consistent and presents the advantage of being computationally much faster than the autoregression bootstrap. For this reason, we are specially interested in analyzing its performance in our clustering algorithm. The three resampling procedures, the autoregression bootstrap (AB), the sieve bootstrap (SB) and the conditional bootstrap (CB), are discussed and compared in the Monte Carlo study of Section 3.

Hereafter, $\boldsymbol{X}_k^{(i)\star j} = \left(X_{T+1}^{(i)\star j}, \ldots, X_{T+k}^{(i)\star j}\right)$ denotes the j-th bootstrap replicate of the predictor $\boldsymbol{X}_k^{(i)} = \left(X_{T+1}^{(i)}, \ldots, X_{T+k}^{(i)}\right)$, for $j = 1, \ldots, B$, $i = 1, \ldots, s$, regardless of the considered bootstrap mechanism.

2.2. Dimension reduction using a PCA-based approach

`Step 2` of the clustering algorithm is aimed at determining a low-dimensional space where all the bootstrap predictors $\boldsymbol{X}_k^{(i)\star j}$ are projected. The objective is simply determining a smaller set of variables that retain as much information from the predictions as possible, and therefore a dimension reduction device is required. Among the available dimension reduction methods (principal component analysis, factor analysis, independent component analysis, ...), we have considered an approach based on principal components analysis (PCA) for several reasons. PCA constructs mutually orthogonal linear combinations of k variables, called principal components, in such a way that a small number p ($p < k$) of these combinations account for most of the variation in the set of original variables. It is the most used dimensionality reduction method due to its simplicity and good properties. Although most inference procedures based on principal components rely on the assumption of independence, PCA may still be performed with dependent data. Furthermore, when the main objective of PCA is descriptive, not inferential, the dependence of the observations does not seriously affect this objective ([18], chapter 12). Several examples applying PCA to time series data can be seen in [18] and references therein. An alternative approach under dependence conditions is factor analysis (FA). However, FA is aimed to model the correlations structure among the original variables and this is not actually our objective. Further, PCA is less sensible to the dimensionality of the model and computationally simpler than common FA. In our particular problem, PCA is conducted as follows.

First, each of the original series $\boldsymbol{X}^{(i)} = \left(X_1^{(i)}, \ldots, X_T^{(i)}\right)$ is split into non-overlapping blocks of k consecutive values. Then, the set S of original series is rearranged to construct the $n \times k$ matrix given by $\boldsymbol{\Lambda} = \left(\boldsymbol{\Lambda}^{(1)}, \boldsymbol{\Lambda}^{(2)}, \ldots, \boldsymbol{\Lambda}^{(s)}\right)^t$, with

$$\boldsymbol{\Lambda}^{(i)} = \begin{pmatrix} X_T^{(i)} & X_{T-1}^{(i)} & \cdots & X_{T-k+1}^{(i)} \\ X_{T-k}^{(i)} & X_{T-k-1}^{(i)} & \cdots & X_{T-2k+1}^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ X_{T-rk}^{(i)} & X_{T-rk-1}^{(i)} & \cdots & X_{T-rk+1}^{(i)} \end{pmatrix}, \tag{2.2}$$

for $i = 1, \ldots, s$, with r being the largest integer such that $rk < T + 1$ and $n = (r + 1)s$.

Matrix $\mathbf{\Lambda}$ is used as the input data matrix to carry out the principal components analysis. As each row of $\mathbf{\Lambda}^{(i)}$ consists of a realization of k consecutive values of $\boldsymbol{X}^{(i)}$, the covariance matrix associated with the columns of $\mathbf{\Lambda}$ estimates the pooled auto-covariance structure for consecutive time lags $1, 2, \ldots, k$. Hence, the principal components are strongly influenced by the first k lagged correlations of the original series, and therefore the components provide additional insight into the dependence structure of the prediction vectors $\boldsymbol{X}_k^{(i)}$. For instance, if the generating processes of the series $\boldsymbol{X}^{(i)}$ and $\boldsymbol{X}^{(j)}$, $i \neq j$, have different covariance structures, then the rows from $\mathbf{\Lambda}^{(i)}$ and $\mathbf{\Lambda}^{(j)}$ will be projected onto different areas of the low-dimensional space defined by the first few components. Such behaviour is certainly desirable to properly classify the prediction vectors.

A small number p of principal components, $\mathcal{C} = \{C_1, \ldots, C_p\}$, accounting for as much of the variability as possible, is then retained. The value of p should lead to a reasonable trade-off between a small number of components and a high explained variance, but its choice must be carefully examined bearing in mind the need of constructing kernel estimators of p-variate densities and the high computational cost of the procedure.

The bootstrap predictions $\boldsymbol{X}_k^{(i)\star j}$ are then projected to the new p-dimensional space determined by the components in $\mathcal{C}$, thus obtaining their scores on this new coordinate system, which are denoted by $\boldsymbol{Z}_p^{(i)\star j} = \left(Z_1^{(i)\star j}, \ldots, Z_p^{(i)\star j}\right)$, for $j = 1, \ldots, B$ and $i = 1, \ldots, s$. These sets of scores are used in the following step of the clustering algorithm to compute kernel-type non-parametric estimators of the unknown p-dimensional densities of $\boldsymbol{Z}_p^{(i)}$, for $i = 1, \ldots, s$. Specifically, we consider the standard p-dimensional kernel density estimator given by

$$\hat{f}_{\boldsymbol{H}}^{(i)}\left(\boldsymbol{z}_p\right) = \frac{1}{B} \sum_{j=1}^{B} K_{p,\boldsymbol{H}}\left(\boldsymbol{z}_p - \boldsymbol{Z}_p^{(i)\star j}\right), \tag{2.3}$$

where $\boldsymbol{H}$ is a symmetric positive definite $p \times p$ matrix (bandwidth matrix) and

$$K_{p,\boldsymbol{H}}\left(\boldsymbol{z}_p\right) = |\boldsymbol{H}|^{-1/2} K\left(\boldsymbol{H}^{-1/2}\boldsymbol{z}_p\right),$$

with K being a p-variate kernel function.

2.3. Computation of an initial dissimilarity matrix

In accordance with our proposal, the distance between two series $\boldsymbol{X}^{(i)}$ and $\boldsymbol{X}^{(j)}$ must assess the discrepancy between the p-variate densities of $\boldsymbol{Z}_p^{(i)}$ and $\boldsymbol{Z}_p^{(j)}$, where $\boldsymbol{Z}_p^{(i)}$ denotes the projection onto the principal components space of the predictor $\boldsymbol{X}_k^{(i)}$. As these p-variate densities are unknown, the chosen metric must be based on their kernel approximations $\hat{f}_{\boldsymbol{H}_i}^{(i)}$ and $\hat{f}_{\boldsymbol{H}_j}^{(j)}$, introduced in (2.3). In particular, two metrics have been considered. First, the L^1 functional distance given by

$$D_{ij}^{(1)} = \int_{\mathbb{R}^p} \left| \hat{f}_{\boldsymbol{H}_i}^{(i)}(\boldsymbol{z}_p) - \hat{f}_{\boldsymbol{H}_j}^{(j)}(\boldsymbol{z}_p) \right| d\boldsymbol{z}_p. \tag{2.4}$$

Although classical multivariate methods are based on the use of the L^2 distance, mainly by its analytical tractability, Vilar et al. [34] have obtained best results with the L^1 distance in a similar clustering setup. They argue that if a pair of densities are very far apart, with disjoint supports, then the L^2 distance removes the effect of the distance between their centroids and it is only governed by the shape of the densities. This feature may substantially distort the initial dissimilarity matrix, and hence yield a poor performance in the clustering task. Unlike the L^2 distance, the L^1 distance between two densities with disjoint supports is exactly equal to two, regardless of the shape of the densities. Thus, $D_{ij}^{(1)}$ allows us to correctly identify the most distant series and leads to a reasonable cluster solution. Also it is well-known that L^1 distance is less sensitive to outliers compared to the L^2 distance.

On the other hand, even though our algorithm is aimed at reducing the dimension of the space where the full multivariate forecast densities are estimated, we have considered an alternative distance based on the univariate marginal densities, thus avoiding the problems derived from the multivariate density estimation. Such a distance is defined by

$$D_{ij}^{(2)} = \int_{\mathbb{R}^p} \left| \prod_{l=1}^{p} \hat{f}_{l,h_{l,i}}^{(i)}(z_l) - \prod_{l=1}^{p} \hat{f}_{l,h_{l,j}}^{(j)}(z_l) \right| dz_1 \ldots dz_p, \tag{2.5}$$

where $\hat{f}_{l,h_{l,i}}^{(i)}$, $i = 1, \ldots, p$, is an univariate kernel estimator of the l-th univariate marginal density of the random vector $\boldsymbol{Z}_p^{(i)}$, constructed with bandwidth $h_{l,i}$. Note that the one-dimensional approach used by distance (2.5) is only useful when the components of $\boldsymbol{Z}_p^{(i)}$ are independent. However, since the principal components are uncorrelated, the independence of the Z_is is ensured when the original series have Gaussian innovations. Furthermore, unlike $D_{ij}^{(1)}$, distance $D_{ij}^{(2)}$ allows us to use a large number p of principal components without increasing the computational effort in a substantial way.

Any of these metrics, $D_{ij}^{(1)}$ or $D_{ij}^{(2)}$, allows us to construct a dissimilarity matrix that can be taken as starting point to perform an agglomerative hierarchical clustering. In this way, clustering will be governed by a dissimilarity measure based on the behavior of the predictions, as we intended. Other standard clustering methods do not satisfy this property. For instance, the k-means algorithm moves each series to the cluster whose centroid is closest (usually in terms of the Euclidean distance), recalculates the cluster centroid and repeats the assignment procedure until no time series is reassigned. Therefore, the k-means does not work with the proposed metrics. Further, it is complex to introduce here the concept of centroid. A centroid would be a kernel prediction density generated from an averaging of different series, and this is not reasonable at all.

Other partitioning procedures, such as the k-medoids algorithm, could be used. However, unlike the partitioning procedures, the hierarchical methods produce a complete set of cluster solutions, ranging from single-member clusters to the one-cluster solution, thus enabling us to analyze how close two nested partitions are. Considering these arguments, we decide to perform hierarchical clustering in our experiments. Several criteria to link two clusters in an intermediate stage of the hierarchical process are available. The average linkage or Ward's methods are usually preferable to others, such as the single linkage and the complete linkage methods, because the former tend to generate clusters with small and similar within-cluster variation and are less affected by the presence of outliers. Nevertheless, as the k-means algorithm, the Ward's method is not well suited to our problem because it is based on Euclidean distances between centroids, and thus distances $D_{ij}^{(1)}$ and $D_{ij}^{(2)}$ would not be used to link clusters. For it, the Ward's method is not considered in our experiments.

3. Simulation study

Some results from a simulation study carried out to analyze the performance of our clustering methodology are shown in the present section. Our first set of experiments was conducted to analyze the accuracy of the bootstrap-based distances introduced in (2.4) and (2.5), and simultaneously to compare the three considered bootstrap methods. Then, a new set of experiments was designed to assess the quality of the cluster solutions obtained with our clustering procedure.

3.1. Analyzing the accuracy of the bootstrap-based distances

Let $S = \left\{\boldsymbol{X}^{(1)}, \dots, \boldsymbol{X}^{(s)}\right\}$ be a set of T-length series subjected to the clustering algorithm proposed in Section 2. According to the notation used so far, k denotes the length of the forecast vectors $\boldsymbol{X}_k^{(i)} = \left(X_{T+1}^{(i)}, \dots, X_{T+k}^{(i)}\right)$ governing the clustering and p is the number of selected principal components. Now, let $f^{(i)}$ and $f_l^{(i)}$, $l = 1, \dots, p$, be the joint density and the l-th univariate marginal density associated with $\boldsymbol{Z}_p^{(i)}$, the random vector of principal component scores for $\boldsymbol{X}_k^{(i)}$. We are particularly interested in studying the behavior of the quantities given by

$$\begin{aligned} d_{i,k,p}^{(1)} &= \int_{\mathbb{R}^p} \left| \hat{f}_{\boldsymbol{H}_i}^{(i)}\left(\boldsymbol{z}_p\right) - f^{(i)}\left(\boldsymbol{z}_p\right) \right| d\boldsymbol{z}_p, & (3.1) \\ d_{i,k,p}^{(2)} &= \int_{\mathbb{R}^p} \left| \prod_{l=1}^{p} \hat{f}_{l,h_{l,i}}^{(i)}\left(z_l\right) - \prod_{l=1}^{p} f_l^{(i)}\left(z_l\right) \right| dz_1 \dots dz_p, & (3.2) \end{aligned}$$

for $i = 1, \dots, s$, where $\hat{f}_{\boldsymbol{H}_i}^{(i)}$ and $\hat{f}_{l,h_{l,i}}^{(i)}$, $l = 1, \dots, p$, are kernel estimators based on the bootstrap samples of $f^{(i)}$ and $f_l^{(i)}$, respectively.

If quantities $d^{(u)}_{i,k,p}$, with $u = 1$ or 2, are close to zero for all $i = 1, \ldots, s$, then the bootstrap-based distances $D^{(u)}_{ij}$, given in (2.4) and (2.5), approximate correctly their theoretical versions, say $\mathcal{D}^{(u)}_{ij}$, where $\mathcal{D}^{(u)}_{ij}$ is constructed as $D^{(u)}_{ij}$ but using the true densities $f^{(i)}_p$ and $f^{(i)}_l$, $l = 1, \ldots, s$. Otherwise, $D^{(u)}_{ij}$ and $\mathcal{D}^{(u)}_{ij}$ should lead to different cluster solutions, thus concluding the inefficacy of our bootstrap-based clustering.

Our first experiments are then conducted to examine the performance with finite samples of quantities $d^{(u)}_{i,k,p}$, $u = 1, 2$. Unfortunately, these quantities are not feasible in practice because $f^{(i)}$ and $f^{(i)}_l$ are unknown. This problem is overcome by considering kernel estimators based on the principal component scores of Monte Carlo forecasts. These estimators will be denoted by $f^{(i),MC}$ and $f^{(i),MC}_l$, and they can be considered as a benchmark in our experiments because Monte Carlo results are based on the true generating model (model's structure, parameters and innovations distribution). In this way, feasible versions of $d^{(u)}_{i,k,p}$ can be obtained by setting

$$d^{(1),\bullet}_{i,k,p} = \int_{\mathbb{R}^p} \left| \hat{f}^{(i),\bullet}_{\boldsymbol{H}_i}(\boldsymbol{z}_p) - f^{(i),MC}(\boldsymbol{z}_p) \right| d\boldsymbol{z}_p, \tag{3.3}$$

$$d^{(2),\bullet}_{i,k,p} = \int_{\mathbb{R}^p} \left| \prod_{l=1}^{p} \hat{f}^{(i),\bullet}_{l,h_{l,i}}(z_l) - \prod_{l=1}^{p} f^{(i),MC}_l(z_l) \right| dz_1 \ldots dz_p, \tag{3.4}$$

with $\bullet$ taking values in $\{SB, AB, CB\}$ according to the bootstrap procedure used to obtain $\hat{f}^{(i),\bullet}_{\boldsymbol{H}_i}$ and $\hat{f}^{(i),\bullet}_{l,h_l}$, namely the sieve bootstrap (SB), the autoregression bootstrap (AB) and the conditional bootstrap (CB).

The main features of our first set of experiments are detailed below. For each replication of the experiment, the dataset S subjected to clustering was formed by one partial realization of length $T = 200$ simulated from each of the autoregressive models enumerated in Table 1.

In all cases, ε_t are i.i.d. zero-mean Gaussian random variables with variance σ^2. Model M1 is an AR(1) process with moderate autocorrelation. Models M2-M6 form an interesting class of parametric nonlinear autoregressive processes. All of them show different nonlinear structures for the conditional mean, moving from weak to strong non-linearity, and thus providing a valuable scenario to perform our algorithm. These models were also considered in [34] and by other authors in previous works.

Table 1
Autoregressive models considered in the simulation study

M1	AR	$X_t = 0.6X_{t-1} + \varepsilon_t$
M2	Bilinear	$X_t = (0.3 - 0.2\varepsilon_{t-1})\, X_{t-1} + 1.0 + \varepsilon_t$
M3	EXPAR	$X_t = \left(0.9 \exp\left(-X^2_{t-1}\right) - 0.6\right) X_{t-1} + 1.0 + \varepsilon_t$
M4	SETAR	$X_t = (0.3X_{t-1} + 1.0)\, I\,(X_{t-1} \geq 0.2) -$ $(0.3X_{t-1} - 1.0)\, I\,(X_{t-1} < 0.2) + \varepsilon_t$
M5	NLAR	$X_t = 0.7\, \lvert X_{t-1} \rvert \,(2 + \lvert X_{t-1} \rvert)^{-1} + \varepsilon_t$
M6	STAR	$X_t = 0.8X_{t-1} - 0.8X_{t-1}\,(1 + \exp(-10X_{t-1}))^{-1} + \varepsilon_t$

Different values for the length of the forecasts vectors were used, namely $k = 3$, $k = 5$ and $k = 10$. Univariate kernel estimators $\hat{f}^{(i),\bullet}_{l,h_l}$ were constructed with bandwidths $h_{i,l}$ obtained by using the plug-in selector by Wand and Jones [38]. A generalization of this selector to the multivariate case was also considered to construct $\boldsymbol{H}_i$, the bandwidth matrix for the multivariate estimator $\hat{f}^{(i),\bullet}_{\boldsymbol{H}_i}$. The number of bootstrap replicates was always $B = 1,000$. Standard PCA based on the covariance matrix was carried out and the number p of retained principal components was fixed to be $p = 2$. The proportion of explained variance with the retained components was recorded in each case. Under these simulation features, each experiment was replicated a total of $N = 200$ times, so that 200 values of $d^{(u),\bullet}_{i,k,p}$ were computed for each $u = 1, 2$ and $\bullet = SB$, AB and CB. Figure 1 shows the boxplots constructed with these values for $k = 5$.

Boxplots in Figure 1 show that the sieve bootstrap (SB) only works well with data generated from the linear model M1, while the autoregression bootstrap

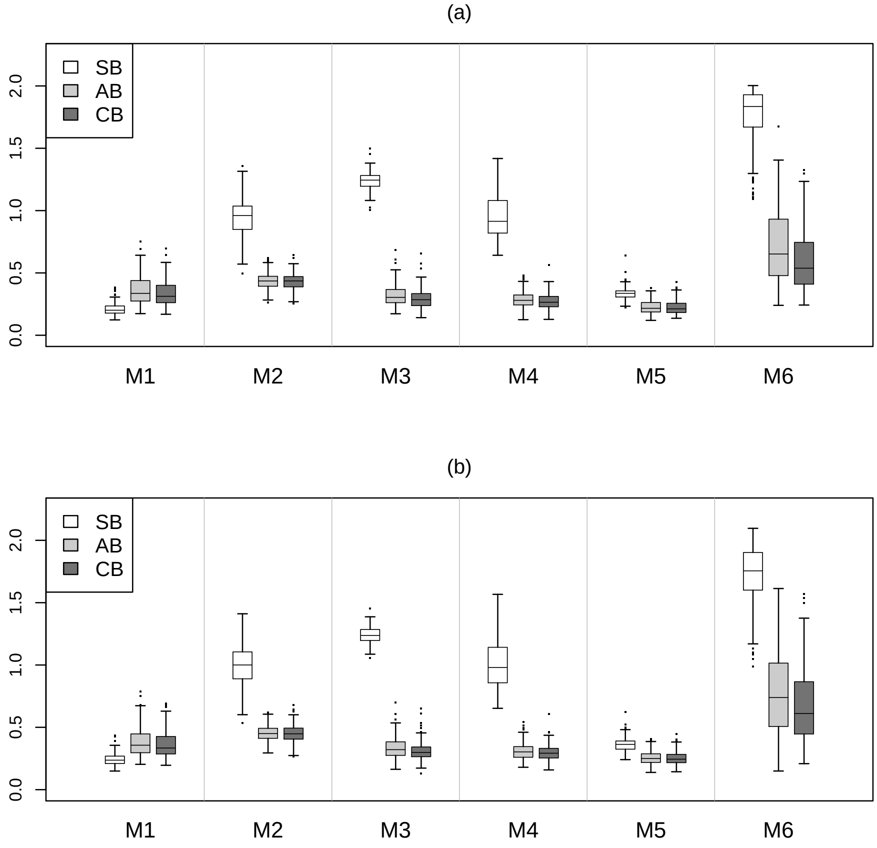

Fig 1. *Boxplots of values* $d^{(u),SB}_{i,5,2}$, $d^{(u),AB}_{i,5,2}$ *and* $d^{(u),CB}_{i,5,2}$ *for (a)* $u = 1$ *(using bidimensional densities) and (b)* $u = 2$ *(using univariate marginal densities).*

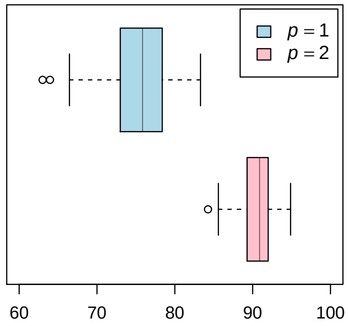

Fig 2. *Boxplots of proportions of cumulative variance explained by the first* ($p = 1$) *and the first two* ($p = 2$) *principal components with* $k = 5$.

(AB) and the conditional bootstrap (CB) clearly outperform the sieve bootstrap when the underlying model presents non-linearity. SB also yields poor results with models presenting a weak non-linear structure, such as M5 and M6 where the underlying processes can be well approximated by linear processes in the short time. Therefore, it is evident that the nonparametric bootstrap methods show a great versatility, thus permitting to extend the range of applicability of the clustering algorithm. On the other hand, similar results are obtained with AB and CB, although a slight improvement seems to be observed with CB. Note that, except for Model M6, where the worst results are observed, both AB and CB lead to very similar results for all the models, which shows that these algorithms are not severely affected by departures from the data generating model.

Comparison of both panels in Figure 1 allows us to conclude that $d^{(1),\bullet}$ and $d^{(2),\bullet}$ yield results close to zero when an appropriate bootstrap procedure is considered. Furthermore, both quantities perform in a very similar way, which is expected because Gaussian innovations were used to simulate all the series.

It is also worth stressing that only $p = 2$ principal components allowing us to explain a high percentage of the original variability. Boxplots of the proportions of cumulative variance explained by the principal components at each trial of the experiment are shown in Figure 2. Both boxplots are located on short ranges of high cumulative variance proportions, thus showing the ability of PCA to achieve a substantial dimension reduction.

Contour diagrams of the kernel density estimators for some arbitrary trials are displayed in Figure 3 to illustrate as the bootstrap forecasts of each series are projected onto the space formed by the first two principal components.

Figure 3 suggests that the series from Model M6 should be easily identified as an isolated group in the clustering because their contour diagrams show a different covariance pattern and centroids located far from the rest of centroids. Densities associated with series generated from Model M3 present a more spheri-

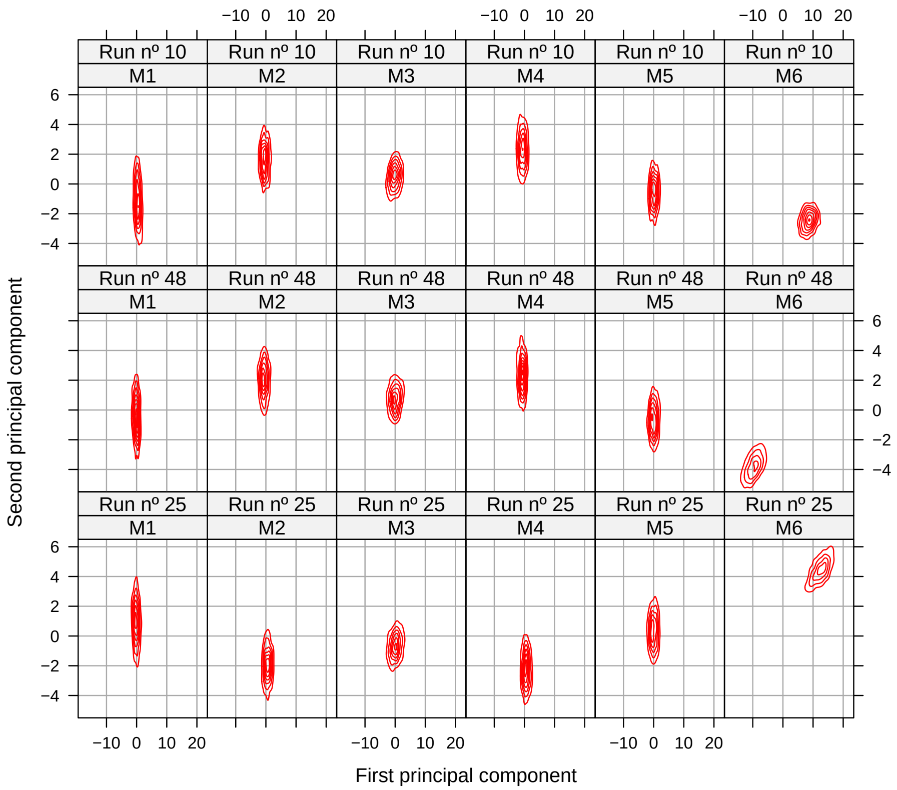

FIG 3. *Contour diagrams of kernel density estimators based on the* $1,000$ *CB-bootstrap forecasts projected onto the principal components space* ($p = 2$) *for three trials of the simulation experiment.*

cal shape, and hence it is expected that these series are grouped forming a cluster that will remain separate from the rest until very late in the hierarchical process. Series from the remaining four models seem to determine two clusters: {M1,M5} and {M2,M4}. In fact, series from models M1 and M5, and analogously those from M2 and M4, show contour diagrams very similar and with high amount of overlapping, thus both clusters should be formed at an early stage of the hierarchical process. Note that the first principal component allows us to separate series from Model M6, while the differences between clusters {M1,M5} and {M2,M4} become evident on the second principal component.

3.2. Evaluating the hierarchical solutions

Once observed that the bootstrap-based distances approximate correctly their corresponding Monte Carlo versions, a new experiment is carried out to examine the quality of the cluster solutions generated with our procedure. The objective is to measure the agreement between the experimental solution and the true cluster partition by using different cluster similarity indexes.

According to our clustering principle, the "true" cluster solution must be constructed taking as starting point the dissimilarity between the forecast densities for the next k future times, say $\overline{f}^{(i)}(\boldsymbol{X}_k^{(i)})$, for $i = 1, \ldots, S$. As these densities are unknown, kernel estimators based on Monte Carlo forecasts $\hat{\overline{f}}^{(i),MC}$ are previously constructed for $i = 1, \ldots, S$. Then, the true cluster solution corresponds to the output of a hierarchical clustering based on the dissimilarity matrix $\overline{D}^{MC}$, whose (i, j)-th element measures the L^1 distance between $\hat{\overline{f}}^{(i),MC}$ and $\hat{\overline{f}}^{(j),MC}$. Basically, the idea is to perform clustering without reducing the dimension of the space where the forecast densities are estimated. Therefore, our experiments in this section must be limited to scenarios with small values of k in order to overcome the adverse impact of the *curse of dimensionality* on the kernel density estimators (this drawback is, in fact, the main motivation to perform PCA in our methodology).

Once the experimental cluster solution (obtained by applying our method) and the true cluster solution are available, we focus on comparing the r-cluster solutions of both hierarchies for different values of r. Three cluster similarity indexes are considered and presented below.

Denote by $\mathcal{T}_r = \{T_1, \ldots, T_r\}$ and $\mathcal{E}_r = \{E_1, \ldots, E_r\}$ the true and experimental r-cluster solutions, respectively. The first considered cluster similarity measure was introduced by Gavrilov et al. [13] and is defined by

$$GI\left(\mathcal{T}_r, \mathcal{E}_r\right) = \frac{1}{r}\sum_{i=1}^{r} \max_{1\leq j\leq r} GI\left(T_i, E_j\right), \tag{3.5}$$

where

$$GI\left(T_i, E_j\right) = \frac{2\left|T_i \cap E_j\right|}{\left|T_i\right| + \left|E_j\right|}$$

and $|\cdot|$ denotes the cardinality of the elements in each set. Note that GI is 0 if both partitions are completely dissimilar and 1 if they are identical.

An alternative index frequently used to measure the agreement between two partitions is the Rand index (RI) [30], which calculates the proportion of the number of pairs of series that are either into the same cluster or into different clusters by both partitions. RI varies in the range $[0, 1]$. The greater RI, the higher is the agreement between both partitions. The third index is the Adjusted Rand index (ARI) [16], a corrected-for-chance version of the Rand index satisfying that its expected value is zero when the partitions are selected at random. ARI lies between -1 and 1 and is exactly 1 when the two partitions agree perfectly.

The new simulation experiment was conducted as follows. At each trial of the experiment, three series of length $T = 200$ were generated from each of models M1 to M6. The resulting set of $S = 18$ series was subjected to hierarchical

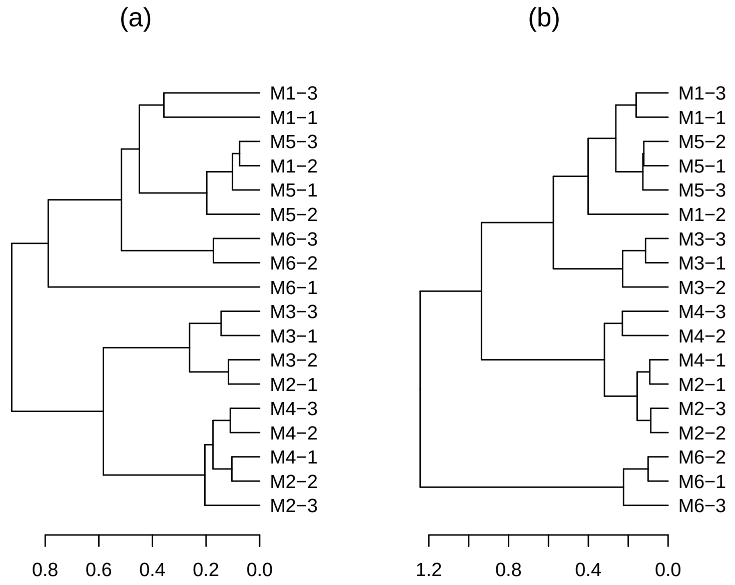

Fig 4. *Dendrograms for one particular trial obtained as follows: (a) without reducing the dimension of the problem and using Monte Carlo replicates, and (b) applying our methodology with CB bootstrap. The average linkage is used as agglomerative method.*

clustering based on grouping series with similar behavior for their first two predictions, i.e. $k = 2$. Two hierarchies of clusters or dendrograms were obtained. Once of them is the experimental hierarchical solution generated by applying our clustering methodology with $p = 1$ and distances $D_{ij}^{(2),CB}$, given in (2.5), with density estimators based on CB predictions. The other cluster solution is the true hierarchical solution obtained by performing clustering without reducing the dimension of the problem, i.e. starting from the L^1 distances between forecast densities estimated with Monte Carlo replicates in the original space of dimension $k = 2$. Both clustering processes are carried out for several measures of proximity between groups, including single linkage, complete linkage and average linkage. The cluster similarity indexes GI, RI and ARI, were computed for partitions of different sizes. This process was replicated a total of $N = 200$ times.

To gain insight into the structure of the clustering solutions, the dendrograms obtained using the average linkage for a particular trial are shown in Figure 4.

The dendrogram generated with our procedure (Figure 4(b)) allows us to identify four reasonably homogeneous clusters formed by series from models M1-M5, M2-M4, M3 and M6. This classification is consistent with our discussion in Section 3.1 for $k = 5$ y $p = 2$. The dendrogram corresponding to the true clustering in Figure 4(a) shows a high concordance level with the experimental

TABLE 2
Averages of the cluster similarity indexes for several partition sizes with different agglomerative algorithms

	r-cluster solution								
	2	3	4	5	6	7	8	9	10
Average Linkage									
GI	0.734	0.727	0.711	0.705	0.693	0.689	0.695	0.710	0.724
RI	0.716	0.737	0.790	0.824	0.837	0.847	0.860	0.876	0.892
ARI	0.397	0.470	0.516	0.517	0.478	0.417	0.370	0.340	0.301
Complete Linkage									
GI	0.777	0.754	0.726	0.704	0.681	0.683	0.693	0.705	0.719
RI	0.719	0.762	0.805	0.825	0.835	0.853	0.868	0.884	0.897
ARI	0.421	0.499	0.515	0.486	0.423	0.387	0.350	0.321	0.282
Single Linkage									
GI	0.673	0.610	0.644	0.670	0.683	0.688	0.697	0.711	0.728
RI	0.706	0.617	0.682	0.754	0.800	0.823	0.843	0.863	0.878
ARI	0.328	0.273	0.368	0.441	0.452	0.430	0.396	0.367	0.321

dendrogram. In this particular case, the cluster similarity indexes for the 4-cluster solutions take the values $GI = 0.780$, $RI = 0.856$ and $ARI = 0.629$. The most significant changes are that M6-1 appears to be isolated point until very late in the agglomerative process and M2-1 is placed in the group of series M3.

To obtain an overall evaluation of the agreement between the outputs from both clustering processes, the cluster similarity indexes were averaged over the 200 trials. The corresponding mean values are provided in Table 2.

Table 2 shows that high agreement levels are achieved with all considered linkage methods and for different partition sizes. For instance, if we focus on the results for the 4- and 5-cluster solutions obtained with the complete and average linkages, then it is observed that Gavrilov and Rand indexes take always values above 0.7 and 0.8, respectively. The Adjusted Rand index is typically lower than the Rand index but, in any case, a high value of 0.515 is achieved for this index. If the single linkage procedure is used, the results are also satisfactory although somewhat worse.

4. Case study: Clustering forecasts of electricity prices

In this section the proposed clustering algorithm is applied to a set of series of electricity prices. Prediction of electricity price is an important issue in competitive electric power markets. If producers and consumers have reliable predictions of electricity price, they can develop their bidding strategies and establish a pool bidding technique to achieve a maximum benefit. However, price series exhibit features that make their analysis difficult (calendar effect on weekends and holidays, outliers in periods of high demand, high volatility . . .). The monograph by Weron [39] provides an interesting survey of modern tools for modeling and forecasting electricity loads and prices. An issue of particular interest is the short term price prediction and, in particular, many references have treated

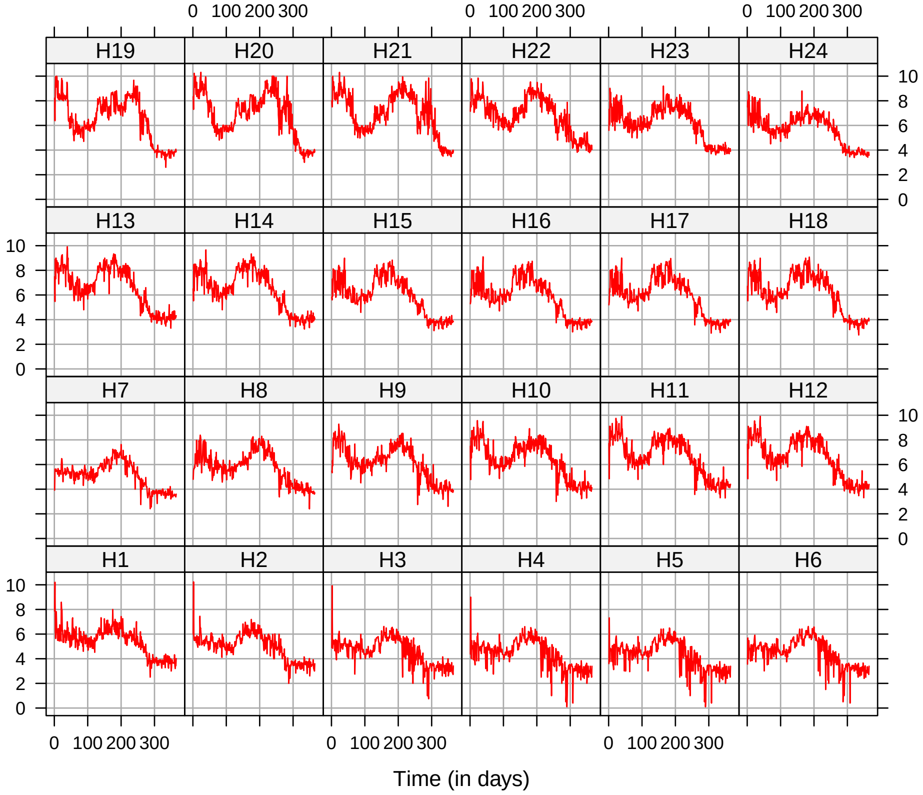

Fig 5. *Daily series (weekends excluded) of electricity price for each hour of the day from December 31, 2007 to May 25, 2009 ($T = 365$ observations).*

with the problem of the one-day-ahead forecasting (see [22, 8, 12, 36], among many others). As the electricity prices vary throughout the time of day, one of the strategies considered to face this problem consists in building 24 models to compute one-day-ahead hourly predictions, using a different time series for each hour. This is the approach that we follow in the present section. In connection with our clustering proposal, our specific interest is to group hours with similar several-days-ahead predictions.

The dataset in study consists of hourly electricity prices in the Spanish market during the period December 31, 2007 - May 25, 2009. Data are available at `http://www.omel.es`, the official website of Operador del Mercado Ibérico de Energía. Records corresponding to Saturdays and Sundays are excluded from the database because electricity demand, and hence electricity price, are lower on the weekends. In this way, we have 24 time series $\boldsymbol{X}^{(i)} = \left(X_1^{(i)}, \ldots, X_T^{(i)}\right)$, $i = 1, \ldots, 24$, of length $T = 365$, where the i-th series provides the daily sequence (weekends excluded) of electricity prices at hour i. A lattice plot of the series in study is given in Figure 5.

Table 3
Component loadings from PCA based on the transformed series

	PC 1	PC 2	PC 3	PC 4	PC 5
1-step-ahead	-0.02543	-0.05906	0.88189	-0.4574	-0.09465
2-step-ahead	0.34408	-0.34448	0.34970	0.7548	-0.26634
3-step-ahead	0.03813	0.76043	0.02965	0.0896	-0.64139
4-step-ahead	-0.53812	-0.49550	-0.18228	-0.1240	-0.64521
5-step-ahead	0.76807	-0.23254	-0.25665	-0.4446	-0.30402

Table 4
Importance of the components derived from the PCA based on the transformed series

	PC 1	PC 2	PC 3	PC 4	PC 5
Standard deviation	0.226	0.189	0.132	0.127	0.063
Proportion of variance	0.411	0.287	0.141	0.129	0.032
Cumulative proportion	0.411	0.698	0.839	0.968	1.000

The 24 series are subjected to our clustering algorithm by taking a horizon from length $k = 5$ days, i.e. series are grouped together whether their predictions for the next five days perform similarly. Note that all series are clearly non-stationary and hence the clustering algorithm cannot be directly applied because of the bootstrap predictions in `Step 1` are computed under stationary assumption. Then we proceeded as follows.

First, each of the time series is transformed using logarithms and taking an appropriate number of regular differences. The software package TRAMO (Time series Regression with ARIMA noise, Missing observations and Outliers) developed by Gómez and Maravall [14] was used to determine the order of regular differences. `Steps 1` and `2` of the procedure are then applied to the transformed series, say $\boldsymbol{Y}^{(i)}$. From `Step 1`, bootstrap prediction vectors of length $k = 5$, $\boldsymbol{Y}_k^{(i)\star j}$, $j = 1, \ldots, B = 1000$, $i = 1, \ldots, 24$, are obtained by using any of the resampling procedures considered in Section 2, namely AB, CB or SB. PCA carried out in `Step 2` leads to a lower-dimensional space where the bootstrap predictors are projected. The component loadings and the amount of variance explained by each component are shown in Tables 3 and 4, respectively. In this case, $p = 2$ or $p = 3$ principal components must be retained to achieve for a reasonable percentage of explained variance.

Now, according to `Step 3`, component scores $\boldsymbol{Z}_p^{(i)\star j}$ generated in `Step 2` should be used to construct the required density estimators. However, the density estimators based on $\boldsymbol{Z}_p^{(i)\star j}$ differ in shape but not in location, since the distances between their centroids have been canceled by working with the transformed series. To correct this undesirable effect, the z-scores are shifted as follows. The 1000 bootstrap predictors of each series are back-transformed to obtain bootstrap predictors for the original series, and the centroid of each resulting group is projected onto the principal components space. If $\boldsymbol{M}_p^{(i)}$ denotes the coordinates in the principal components space of the i-th centroid, then scores $\boldsymbol{Z}_p^{(i)\star j}$ are shifted by setting $\boldsymbol{Z}_p^{(i)\star j} + \boldsymbol{M}_p^{(i)}$, for $i = 1, \ldots, 24$. Vectors $\boldsymbol{M}_p^{(i)}$ and scatter-plots of shifted scores $\boldsymbol{Z}_p^{(i)\star j} + \boldsymbol{M}_p^{(i)}$ are depicted in Figures 6

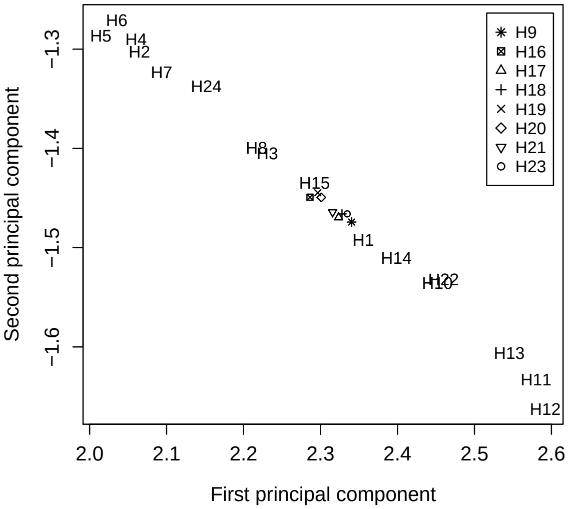

FIG 6. *Centroids of each group of back-transformed bootstrap predictors projected onto the first two principal components. Results when the conditioned bootstrap procedure is used.*

and 7 respectively, for the case where $p = 2$ and the conditioned bootstrap are considered.

Shifted scores $\boldsymbol{Z}_p^{(i)\star j} + \boldsymbol{M}_p^{(i)}$ are then used to carry out `Step 3` of the clustering algorithm, i.e. the dissimilarity between two series is measured by the distance between the kernel approximations to the densities of their corresponding shifted component scores. The rest of steps of the clustering algorithm are then completed as were described in previous sections. Figure 8 shows the resulting dendrogram when $p = 2$ principal components are considered and the clustering is carried out with the average linkage method.

The dendrogram in Figure 8 provides a sequence of nested cluster solutions and the appropriate partition must be determined. Two selection criteria based on choosing the partition that maximizes the value of a specific validation statistic are considered. Both criteria are described below.

Average Silhouette Width (ASW) (Kaufman and Rousseeuw [21]) For a particular partition of S objects into clusters $G_1, \ldots, G_r$, the silhouette width $s_r(i)$ for each object i is defined by

$$s_r(i) = \frac{b_r(i) - a_r(i)}{\max\{a_r(i), b_r(i)\}}$$

where

$$a_r(i) = \frac{1}{|G_s| - 1} \sum_{j \in G_s, j \neq i} d_{i,j}$$

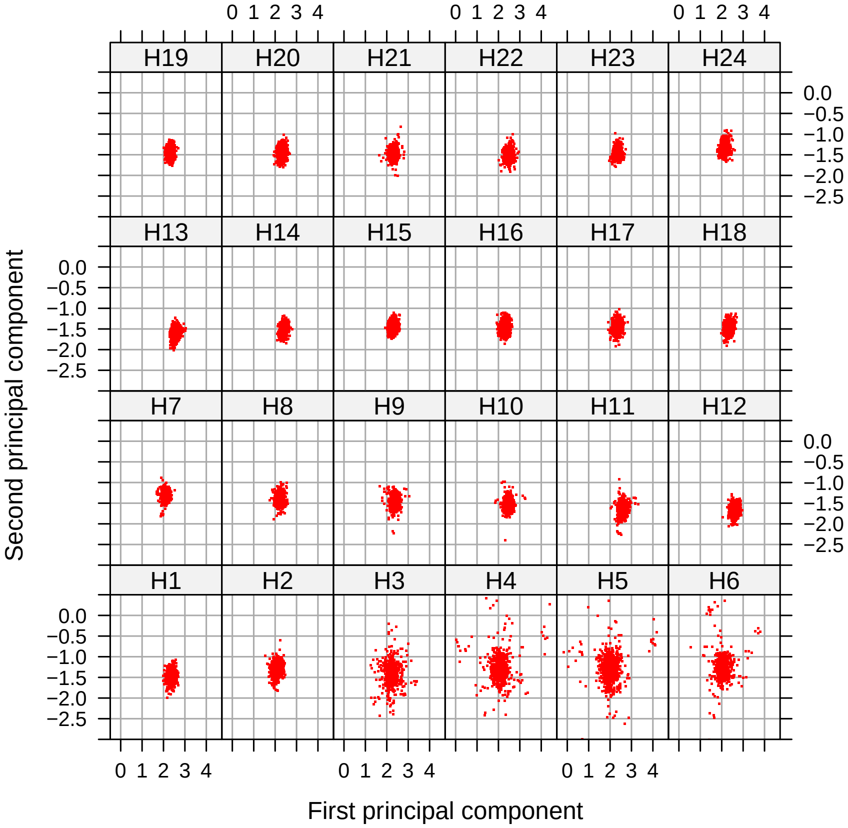

Fig 7. *Scatter plots of shifted scores* $\boldsymbol{Z}_p^{(i)\star j} + \boldsymbol{M}_p^{(i)}$. *Results when the conditioned bootstrap procedure is used.*

is the average dissimilarity between i and all other objects of the cluster G_s to which i has been assigned, and

$$b_r(i) = \min_{l \neq s} \frac{1}{|G_l|} \sum_{j \in G_l} d_{i,j}$$

is the average dissimilarity between i and all objects in the closest cluster G_u, $u \neq s$, i.e. G_u is the second-best choice for object i. The silhouette width always takes values between -1 and 1 and admits a simple interpretation: objects with $s_r(i)$ close to one are very well clustered, a small $s_r(i)$ (around 0) means that the object lies between two clusters, and objects with a $s_r(i)$ close to -1 are probably placed in the wrong cluster.
The average silhouette width, $\text{ASW}(r) = \frac{1}{S}\sum_{i=1}^{S} s(i,r)$, provides an overall measure of clustering performance and a useful criterion for assessing

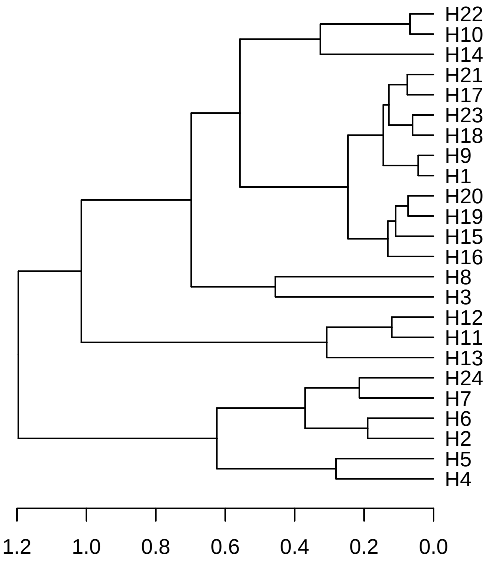

FIG 8. *Dendrogram based on data depicted in Figure 7 by using the average linkage algorithm as proximity measure between groups.*

the number of clusters by choosing the partition that maximizes ASW(r), for $r = 2, \dots, S - 1$.

Pearson version of Hubert's Γ (PH) (Jain and Dubes [17]) For a partition with r groups, the PH statistic is given by the Pearson correlation $\rho(\boldsymbol{d}, \boldsymbol{m}(r))$ between the vector of pairwise dissimilarities $\boldsymbol{d}$ and the binary vector $\boldsymbol{m}(r)$ that is 0 for every pair of observations in the same cluster and 1 for every pair of observations in different clusters. PH measures, in some sense, how good the clustering is as an approximation of the dissimilarity matrix. As before, the objective is to determine the partition maximizing PH(r), for $r = 2, \dots, S - 1$.

Figure 9 shows the values of statistics ASW and PH as function of the number of clusters for the dendrogram in Figure 8.

Both criteria lead to select the 3-cluster solution, with clusters formed by $G_2 = \{H2, H4, H5, H6, H7, H24\}$, $G_3 = \{H11, H12, H13\}$ and G_1 grouping the rest of series. The average silhouette width for $r = 3$ is 0.554, which suggests a reasonably accurate clustering. In fact, the three clusters have an average silhouette width greater than 0.5, and only five series in G_1, namely $H3$, $H8$, $H10$, $H14$ and $H22$, present individual silhouette widths close to zero (below 0.3) and could be not properly classified (see silhouette plot in Figure 10(a)). More specifically, $H3$ and $H8$ lie very close to cluster G_2 and $H10$, $H14$ and

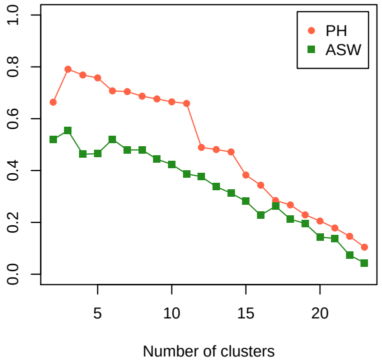

Fig 9. *Values of statistics ASW and PH as function of the number of clusters for the dendrogram in Figure 8.*

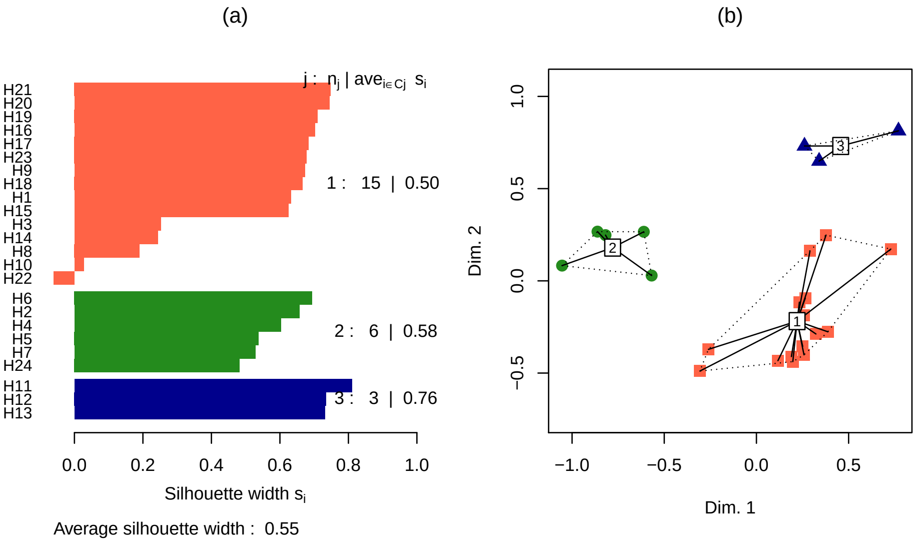

Fig 10. *Two-dimensional representation of the locations of the series based on classical multidimensional scaling of the dissimilarity matrix.*

$H22$ are close to G_3. Classical multidimensional scaling (MDS) [27] of the dissimilarity matrix based on two dimensions was also performed to obtain a visual representation of the proximity among the series. Figure 10(b) shows the plot provided by MDS, and, in addition, the points representing each series are connected to the corresponding cluster centroid with segments. It is observed that

Table 5
Some computing times (in hours) as a function of the number of series subjected to clustering

	Number of series			
	10	20	50	100
Computing time (in hours)	0.21	0.87	6.17	22.01

the three clusters formed are well separated. Clusters G_2 and G_3 are clearly compacts, but even G_1 seems to be reasonably stable as well. The obtained 3-cluster solution is consistent with the location of centroids (Figure 6) and the low dispersion of the scatter-plots (Figure 7). In general, distances between centroids determine the clustering but not in all cases. For instance, $H4$ is closest to $H2$ than to $H5$ according to the location of their centroids (see Figure 6), however $H4$ and $H5$ are grouped together in an earlier stage due to the similarity between their densities (see Figure 7).

Finally, it is worth stressing that our procedure has a high computational cost because of each distance between a pair of series involves the calculation of a multiple integral. To provide accurate information to the reader on this point, we have measured the computing times required for our procedure as a function of the number of series. Our clustering algorithm was run on a PC with the system specifications given by: Intel Core I7 - 2600 processor, 3.46 to 3.7 Ghz CPU, 32 GB HDD, 24 GB of RAM, Windows XP. Series of length $T = 200$, $k = 5$, 2 principal components and the distance $D_{ij}^{(2)}$ given in (2.5) were always considered. The double integrals were estimated using a Monte Carlo algorithm for multidimensional numerical integration and, in addition, integrands were previously approximated by bivariate interpolation onto a 15 × 15 grid of the integration domain. All our code was implemented in the R language [33] and it is available upon request. Results are shown in Table 5.

5. Concluding remarks

The problem of clustering time series is studied for a general class of autoregressive models. A clustering procedure aimed to group series with similar forecasts for a specific sequence of future times is proposed. Specifically, the proposed procedure evaluates the affinity between two series in terms of the distance between their multidimensional forecast densities. The idea is to approximate the forecast densities by using kernel density estimators based on bootstrap replications of the prediction vectors. However, to circumvent the high dimensionality problem, these densities are constructed in a lower-dimensional space where the bootstrap predictors are previously projected. The transformed space is determined by following a PCA-based approach. Three resampling procedures and two different metrics are examined and compared in a simulation study, where a wide variety of autoregressive models is considered. Simulation results show the good behavior of the two nonparametric bootstrap methods in all situations, that is with linear and nonlinear autoregressive models. The approach proposed to develop PCA also yields satisfactory results since an optimal dimension reduction is attained and the experimental hierarchical cluster solutions are consistent with those based on Monte Carlo (the benchmark in our numerical study). The usefulness of the clustering methodology is illustrated through an application to a real data set involving electricity prices series. In this particular case, slight modifications of the clustering procedure are required because the data set is formed by non stationary time series.

An interesting issue to address in future research is to extend this clustering methodology to the case of more complex dynamic models, which would require considering alternative resampling techniques.

Acknowledgements

We thank the two anonymous reviewers and the Associate Editor for their helpful comments.

References

[1] Alonso, A. M., Berrendero, J. R., Hernandez, A. and Justel, A. (2006). Time series clustering based on forecast densities. *Comput. Statist. Data Anal.* **51** 762–776. MR2297485

[2] Alonso, A. M., Peña, D. and Romo, J. (2002). Forecasting time series with sieve bootstrap. *J. Statist. Plann. Inference* **100** 1–11. MR1869807

[3] Boets, J., De Cock, K., Espinoza, M. and De Moor, B. (2005). Clustering time series, subspace identification and cepstral distances. *Commun. Inf. Syst.* **5** 69–96. MR2199724

[4] Bühlmann, P. (1997). Sieve bootstrap for time series. *Bernouilli* **3** 123–148. MR1466304 (99c:62243)

[5] Caiado, J., Crato, N. and Peña, D. (2006). A periodogram-based metric for time series classification. *Comput. Statist. Data Anal.* **50** 2668–2684. MR2227325

[6] Cao, R., Febrero-Bande, M., Gonzalez-Manteiga, W., Prada-Sánchez, J. M. and García-Jurado, I. (1997). Saving computer time in constructing consistent bootstrap prediction intervals for autoregressive processes. *Commun. Stat., Simulation Comput.* **26** 961–978. MR2227325

[7] Chouakria-Douzal, A. and Nagabhushan, P. N. (2007). Adaptive dissimilarity index for measuring time series proximity. *Adv. Data Anal. Classif.* **1** 5–21. MR2329160

[8] Conejo, A. J., Plazas, M. A., Espínola, R. and B., M. (2005). Day-ahead electricity price forecasting using the wavelet transformand ARIMA models. *IEEE Trans. Power Syst.* **20** 1035–1042.

[9] Corduas, M. and Piccolo, D. (2008). Time series clustering and classification by the autoregressive metric. *Comput. Statist. Data Anal.* **52** 1860–1872. MR2418476

[10] Franke, J., Kreiss, J.-P. and Mammen, E. (2002). Bootstrap of kernel smoothing in nonlinear time series. *Bernouilli* **8** 1–37. MR1884156 (2002k:62112)

[11] Galeano, P. and Peña, D. (2000). Multivariate analysis in vector time series. *Resenhas* **4** 383–403. MR1844724 (2002g:62129)

[12] García-Martos, C., Rodríguez, J. and Sánchez, M. J. (2007). Mixed models for short-run forecasting of electricity prices: Application for the Spanish market. *IEEE Trans. Power Syst.* **22** 544–552.

[13] Gavrilov, M., Anguelov, D., Indyk, P. and Motwani, R. (2000). Mining the stock market (extended abstract): which measure is best? In *Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD'00* 487–496. ACM, New York, USA.

[14] Gómez, V. and Maravall, A. (1996). Programs TRAMO (Times Series Regression with ARIMA noise, Missing observations and Outliers) and SEATS (Signal Extraction in ARIMA Time Series). Instructions for the user. Working paper 9628, Bank of Spain, Madrid.

[15] Hart, J. D. (1994). Automated Kernel Smoothing of Dependent Data by Using Time Series Cross- Validation. *Journal of the Royal Statistical Society. Series B (Methodological)* **56** 529–542.

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

[17] Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[18] Jolliffe, I. T. (2002). *Principal component analysis*, second ed. *Springer Series in Statistics*. Springer-Verlag, New York. MR2036084 (2004k:62010)

[19] Kakizawa, Y., Shumway, R. H. and Taniguchi, M. (1998). Discrimination and clustering for multivariate time series. *J. Amer. Statist. Assoc.* **93** 328–340. MR1614589

[20] Kalpakis, K., Gada, D. and Puttagunta, V. (2001). Distance measures for effective clustering of ARIMA time-series. In *Proceedings 2001 IEEE International Conference on Data Mining* (N. Cercone, T. Y. Lin and X. Wu, eds.) 273–280. IEEE Comput. Soc.

[21] Kaufman, L. and Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis*. John Wiley and Sons, New York.

[22] Kim, C.-I., Yu, I.-K. and Song, Y. H. (2002). Prediction of system marginal price of electricity using wavelet transform analysis. *Energy Conv. Manag.* **43** 1839 - 1851.

[23] Kovačić, Z. J. (1998). Classification of time series with applications to the leading indicator selection. In *Data science, classification, and related methods. Proceedings of the fifth Conference of the International Federation of Classification Societies (IFCS-96), Kobe, Japan, March 27-30, 1996* 204–207. Springer.

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

[25] Maharaj, E. A. (1996). A significance test for classifying ARMA models. *J. Statist. Comput. Simulation* **54** 305–331. MR1701220

[26] Maharaj, E. A. (2002). Comparison of non-stationary time series in the frequency domain. *Comput. Statist. Data Anal.* **40** 131–141. MR1930467

[27] Mardia, K. V. (1978). Some properties of classical multi-dimensional scaling. *Comm. Statist. A—Theory Methods* **7** 1233–1241. MR80c:62075

[28] Pértega, S. and Vilar, J. A. (2010). Comparing Several Parametric and Nonparametric Approaches to Time Series Clustering: A Simulation Study. *J. Classification* **27** 333–362. MR2748988 (2011m:62215)

[29] Piccolo, D. (1990). A distance measure for classifying ARIMA models. *J. Time Series Anal.* **11** 153–164.

[30] Rand, W. M. (1971). Objective Criteria for the Evaluation of Clustering Methods. *J. Amer. Statist. Assoc.* **66** 846–850.

[31] Samé, A., Chamroukhi, F., Govaert, G. and Aknin, P. (2011). Model-based clustering and segmentation of time series with changes in regime. *Adv. Data Anal. Classif.* **5** 301–321. MR2860103 (2012j:62243)

[32] Struzik, Z. R. and Siebes, A. (1999). The Haar wavelet in the time series similarity paradigm. In *Principles of Data Mining and Knowledge Discovery. Proceedings of the third European Conference, PKDD'99, Prague, Czech Republic, September 15-18, 1999* 12–22. Springer.

[33] R Core Team (2012). R: A Language and Environment for Statistical Computing R Foundation for Statistical Computing, Vienna, Austria ISBN 3-900051-07-0.

[34] Vilar, J. A., Alonso, A. M. and Vilar, J. M. (2010). Non-linear time series clustering based on non-parametric forecast densities. *Comput. Statist. Data Anal.* **54** 2850–2865. MR2720480

[35] Vilar, J. A. and Pértega, S. (2004). Discriminant and cluster analysis for Gaussian stationary processes: local linear fitting approach. *J. Nonparametr. Stat.* **16** 443-462. MR2073035 (2005h:62172)

[36] Vilar, J. M., Cao, R. and Aneiros, G. (2012). Forecasting next-day electricity demand and price using nonparametric functional methods. *Int. J. Electr. Power Energy Syst.* **39** 48-55.

[37] Vilar, J. M., Vilar, J. A. and Pértega, S. (2009). Classifying Time Series Data: A Nonparametric Approach. *J. Classification* **26** 3–28. MR2507823

[38] Wand, M. P. and Jones, M. C. (1994). Multivariate plug-in bandwidth selection. *Comput. Statist.* **9** 97–116. MR1280754

[39] Weron, R. (2006). *Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach. HSC Books.* Hugo Steinhaus Center, Wroclaw University of Technology.