NOTE: As of 28 May 2013 the **statTSclust** package referenced in this thesis is available on CRAN with the name **TSclust**. The package is in continuous development, changes in the API have been made, but all the functionalities are still there.

A package for stationary time series clustering

UNIVERSIDADE DA CORUÑA

Pablo Montero Manso

A thesis submitted for the degree of

Máster de Técnicas Estadísticas

2013

A package for stationary time series clustering

UNIVERSIDADE DA CORUÑA

Pablo Montero Manso

A thesis submitted for the degree of

Máster de Técnicas Estadísticas

2013

El presente documento que tiene como título *A package for stationaty time series clustering* recoge el trabajo realizado por Pablo Montero Manso como Proyecto Fin de Máster de Técnicas Estadísticas bajo la dirección de José Antonio Vilar Fernández.

Fdo.: Pablo Montero Manso   Fdo.: José Antonio Vilar Fernández

A Coruña, 11 de Enero de 2013

Abstract

An **R** package for stationary time series clustering is presented. Its aim is to provide the **R** community with the implementation of well-established peer-reviewed time series dissimilarity measures and specific clustering methods for research and other purposes. This work has been motivated by the inexistence of a previous packages targeting the problem and the certainty of its demand. Special emphasis in the interoperability with existing general purpose clustering methods has been made, along with the ease of use. All dissimilarity functions are accessible individually for an easier extension and use out of the clustering context.

Contents

1 Introduction **1**

2 Dissimilarity measures **5**

2.1 Approaches for measuring dissimilarity between two time series . . . 5

2.2 Free model approaches 6

2.2.1 Minkowski distance 6

2.2.2 Fréchet distance 7

2.2.3 Dynamic Time Warping distance 7

2.2.4 Chouakria-Douzal dissimilarity 8

2.2.5 Autocorrelation-based distance 9

2.2.6 Periodogram-based distances 10

2.2.7 Dissimilarity measures based on nonparametric spectral estimators 12

2.3 Model-based approaches 14

2.3.1 Piccolo distance 14

2.3.2 Maharaj distance 15

2.3.3 Maharaj extended distance 15

2.3.4 Cepstral-based distance 17

2.4 Prediction-based approaches 17

2.4.1 Distance procedure 19

3 Clustering **21**

3.1 Maharaj Clustering Algorithm 22

3.2 Clustering results evaluation criteria 22

3.2.1 Known Ground-truth cluster evaluation criteria 22

3.2.1.1 Gavrilov similarity 22

3.2.1.2 Rand index 23

3.2.1.3 Adjusted Rand index 23

3.2.2 Unknown Ground-truth cluster evaluation criteria 24
3.2.2.1 Silhouette width 24

4 Simulation study 26
4.1 Classification of time series as Stationary or Non-stationary 26
4.2 Classification of Non-linear time series 28

5 Clustering Real Datasets 31
5.1 Population Dataset 31
5.2 Electrocardiogram Dataset 33
5.3 Forecast 34
5.3.1 CO_2 emissions dataset 35
5.3.2 Industrial production indices dataset 36

6 Conclusions 44
6.1 Design discussion 44
6.2 Usage and Functions Description 44
6.3 Conclusions and Future Work 45

Bibliography 68

List of Figures

2.1 Weighting effect based on several values of k. 9

2.2 Three different cluster solutions depending on whether modelling, present information or future values are used. 18

2.3 Considering only point forecast can produce incorrect results. 18

3.1 Maharaj clustering algorithm for time series. 25

5.1 Example of time series for a state in the exponential (California) cluster and a state in the stabilizing cluster (North Dakota). 32

5.2 Result of clustering real data population series using the d_{ACFG} distance measure. 34

5.3 Samples of ECG readings 34

5.4 Result of clustering the ECG series using the d_{ACFG} distance measure. 36

5.5 Annual CO_2 emissions. 38

5.6 Dendogram based on the differences of forecasts densities of CO_2 emissions. 39

5.7 Forecasts densities of CO_2 emissions. 40

5.8 IPC indices. 41

5.9 Dendogram based on the differences of forecasts densities of IPC indices. 42

5.10 Forecasts densities of IPC indices. 43

Chapter 1

Introduction

Clustering is an unsupervised learning task aimed at classifying a set of unlabeled data objects into homogeneous groups in such a way that the within-group-series similarity is minimized and the between-group-series dissimilarity is maximized. Cluster analysis has an enormous range of applications and is one of the multivariate techniques more widely used. Sometimes the clustering task must be performed with particularly complex data type, which generally requires clustering techniques more sophisticated than the conventional ones for multidimensional data points. This is the case of data having the form of time series. 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. However, the conventional clustering methods usually consider proximity measures that 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. On the other hand, 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. Finding stocks that behave in a similar way, determining products with similar selling patterns, identifying countries with similar population growth or regions with similar temperature . . . are some specific applications where the similarity searching among time series is important.

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 (Liao, 2005), 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.

Some illustrative examples reported in the literature are: classification of children on the basis of similarities in behavior through time (Hirsch and DuBois, 1991), comparison of seismological data as the case of distinguishing between earthquake and nuclear explosions waveforms (Kakizawa et al., 1998), organization of industrial data according to average wage rates (Galbraith and Jiaqing, 1999), cluster models of ecological dynamics (Li et al., 2001), clustering of industrialized countries according to historical data of CO_2 emissions (Alonso et al., 2006), detection of similar immune response behaviors of CD4+ cell number progression over patients affected by immune deficiency virus (HIV) (Chouakria-Douzal and Nagabhushan, 2007), clustering of banks on the basis of their weekly share price series (Vilar et al., 2009), clustering of industrial production indices (Vilar et al., 2010), the automatic identification of groups of rail switching operations by analyzing time series of electrical power consumption acquired during these operations (Samé et al., 2011), and many others.

As previously mentioned, one key point in cluster analysis is to determine a similarity or dissimilarity measure between two data objects. In the specific context of time series, the concept of dissimilarity can be approached from many different points of view. A motivating work by Tong and Dabas (Tong and Dabas, 1990) shows how various measures of similarity and dissimilarity among time series lead to unexpected and substantially different cluster solutions. Furthermore, some usual dissimilarity measures could not work adequately with time series. For instance, the Euclidean distance, one of the most common distance measures in clustering, has the important limitation of being invariant to transformations that modify the order of observations over the time (Galeano and Peña, 2000), and therefore it does not take into account the correlation structure of the time series. So, the problem of measuring the degree of affinity between two time series is a key problem in time series clustering.

Corduas and Piccolo (Corduas and Piccolo, 2008) (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 (Kovačić, 1998; Struzik and Siebes, 1999; Galeano and Peña, 2000; Caiado et al., 2006; Chouakria-Douzal and Nagabhushan, 2007), among others). An alternative approach is to assess the discrepancy between the underlying generating processes (some references following this approach are (Piccolo, 1990; Maharaj, 1996, 2002; Kakizawa et al., 1998; Vilar and Pértega, 2004), among many others). Sometimes, the dissimilarity measures are tailored for the problem at hand, emphasizing properties of the time series that are of interest for the specific context. For instance, 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 a pre-specified future time. Works by (Alonso et al., 2006) and (Vilar et al., 2010) focused on this idea and considered a notion of dissimilarity governed by the performance of future forecasts. Specifically, two time series are similar if their forecasts for a specific future time are close. Summing up, there exist a broad range of metrics to compare time series and 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.

Once the dissimilarity measure is determined, an initial pairwise dissimilarity matrix can be computed and several clustering algorithms be then used to form groups of objects. In fact, most of the time series clustering approaches reviewed by Liao (Liao, 2005) are variations of general procedures (e.g. a k-means or a hierarchical clustering) that use non-conventional metrics specifically designed to deal with time series. According to this observation, we have focused on the implementation of a wide range of dissimilarity measures for time series (Chapter 2). Package `statTSclust` includes model-based metrics, free-model metrics and the prediction-based metrics introduced by (Alonso et al., 2006) and (Vilar et al., 2010). Some of these metrics work in the domain time but others are developed in the frequency domain, and it can be observed that a range of statistical techniques (AR models estimation, kernel density estimation, local polynomial regression, automatic bandwidth selectors, resampling procedures, . . .) must be used to construct these metrics. It is worth stressing that the stationarity assumption is required for some of the considered metrics, although others can be also applied to non-stationary series (in fact, an experimental study addressed to classify series as stationary or as non-stationary is developed in Chapter 4). This limitation is inherent to the own metrics, which are only well-defined under this assumption. Occasionally, some particular metrics are not suitable for being merged with classical clustering algorithms, and then specific clustering methods must be provided. For instance, a specific clustering algorithm developed to work with the metric based on p-values proposed by (Maharaj, 2000) is also implemented in `statTSclust` (see Chapter 3 for details). The design philosophy underlying `statTSclust` aims to provide a complete tool to perform clustering of time series, and for this reason the package includes: routines to evaluate the accuracy of the cluster solutions, such as the Gavrilov index (Gavrilov et al., 2000), the Rand index (Rand, 1971) and an adjusted version of the Rand index (Hubert and Arabie, 1985); statistics to measure the clustering quality indicating jointly the compactness of each cluster and the separateness of different clusters, such as the Silhouette coefficients (Kaufman and Rousseeuw, 1990), and some graphical devices to illustrate the clustering performance (e.g dendrograms, Sihouette plots,. . .). The code of some of these routines (as the case of the Gavrilov's index) was specifically developed here and others are available from several R packages (which are loaded when the package is installed). Finally, some simulation results are presented in the present work to illustrate as these metrics perform and some real data examples are also considered to emphasize the applicability of the time series clustering.

Chapter 2

Dissimilarity measures

2.1 Approaches for measuring dissimilarity between two time series

As mentioned in Introduction, differing from traditional static data as points in high dimensional space, time series encode temporal dynamics that are often key in grouping sequences in many real cases. The different approaches for similarity searching in time series database differ mainly because of their notion of similarity/dissimilarity between two time series. Many different measures of pairwise dissimilarity have been proposed in the classification literature and an important number of them are considered in `statTSclust`. Specifically, we have grouped these dissimilarity measures into three categories: model-based metrics, free model metrics and the prediction-based metrics.

The model-based metrics assume some specific form of the underlying generating models and establish the dissimilarity notion in terms of the discrepancy between the corresponding fitted models. The main approach of statistics researchers has been to assume that the underlying models are generated by an ARIMA process, although researchers in speech recognition and machine learning have also adopted alternative models as Markov chains (MC) (see (Ramoni et al., 2002)) or hidden Markov models (HMM) (see (Smyth et al., 1997) and (Oates et al., 1999), among others). One key difference in clustering technique between the ARIMA and the MC/HMM models is that the ARIMA-based approach fits a model to each series before clustering, whereas most research into MC/HMMs involves estimating the HMMs models for each cluster on each iteration of the clustering algorithm. In `statTSclust` the most popular approaches based on ARIMA models are only considered and they are presented in Section 2.3 of the present work.

The model free metrics do not assume an explicit model form for the underlying structures and they are based on directly comparing the original observations (raw-data-based approaches) or specific features extracted from raw data such as autocorrelations or partial autocorrelations (feature-based approaches). The feature-based approaches present some obvious advantages. For one thing, compared to model-based approaches, feature-based approaches are usually easier to implement as most of the features can be obtained in a straightforward manner. For another, rather than using the raw data, clustering time series using a suitable dissimilarity measure based on their features is much more common. The reason is that if the time series values are used, clustering methods will suffer from extremely high dimensional input, especially for those time series collected at fast sampling rates or with large lengths. Consequently, the clustering process will be time consuming and the computational cost of the clustering process will be increased. In that case, it is much more efficient to use particular features of time series in clustering algorithms. A range of model free metrics are implemented in `statTSclust` and they are described in Section 2.2.

Finally, we have also implemented some metrics to perform clustering when the final purpose is not grouping the underlying structures but measuring the similarity between forecasts at a specific future time. These metrics were proposed by by (Alonso et al., 2006) and (Vilar et al., 2010) and they are presented in Section 3.3.

Hereafter, unless otherwise specified, the metrics work with two time series X_t and Y_t of length T (although this limitation can be omitted in some cases).

2.2 Free model approaches

In this section, dissimilarity measures between two time series constructed with assuming specific generating models are introduced. They directly measure the distance between values of the series or between extracted features and all of them have been considered in time series clustering literature.

2.2.1 Minkowski distance

The simplest approach is to treat the time series as an T-dimensional vector and use the L_q Minkowski distance given by

$$d_{MIK}(X_t, Y_t) = \left(\sum_{k=1}^{T} (X_k - Y_k)^q \right)^{1/q}$$

with q a positive integer.

The L_q distance presents serious drawbacks. First, it depends on the scale of data. On the other hand, the closeness between two series depends on the closeness of the values observed at corresponding points of time, regardless of the serial correlation structure. Thus, observations are treated as if they were independent so that, in particular, d_{MIK} is invariant to permutations over time. Therefore, this metric cannot be considered a good measure of dissimilarity between time series data.

2.2.2 Fréchet distance

This distance was introduced by (Fréchet, 1906) to measure proximity between continuous curves, but has been extensively used on the discrete case (see (Eiter and Mannila, 1994)) and in the time series context. A popular intuitive definition of this distance is the minimum length of a leash required to connect a dog with its owner, each one walking along different curves from one endpoint to the other. Both may vary their speed or even stop, but they can never backtrack.

A formal definition for the discrete case can be given as follows. Let the mapping $r \in M$ between time series $X_t = (x_1, ..., x_p)$ and $Y_t = (y_1, ..., y_q)$ be defined as a sequence of m pairs preserving the observation order

$$r = ((x_{a_1}, y_{b_1}), ..., (x_{a_m}, y_{b_m})),$$

with $a_i \in \{1, ..., p\}$, $b_j \in \{1, .., q\}$, and satisfying for $i \in \{1, .., m-1\}$ the following constraints: $a_1 = 1$, $a_m = p$, $a_{i+1} = (a_i$ or $a_i + 1)$ and $b_1 = 1$, $b_m = q$ and $b_{i+1} = (b_i$ or $b_i + 1)$. The Fréchet distance is defined by

$$d_{FRECH}(X_t, Y_t) = \min_{r \in M} |r| = \min_{r \in M} \left(\max_{i=1,..,m} |x_{a_i} - y_{b_i}| \right).$$

2.2.3 Dynamic Time Warping distance

The dynamic time warping distance was studied in (Sankoff and Kruskal, 1983). It can be considered a variant of the Fréchet distance with the length of the mapping $|r|$ defined as

$$|r| = \sum_{i=1,..,m} |x_{a_i} - y_{b_i}|.$$

Hence, the definition of the dynamic time warping distance is:

$$d_{DTW}(X_t, Y_t) = \min_{r \in M} |r| = \min_{r \in M} \left(\sum_{i=1,..,m} |x_{a_i} - y_{b_i}| \right).$$

According to d_{DTW}, two series are similar if there exists a mapping between their observations, expressing a time distortion by an acceleration/deceleration so that the sum of the spans between all couple observations is close. As in the case of d_{MIK}, both d_{FRECH} and d_{DTW} ignore the temporal structure of the values as the proximity is based on the differences $|x_{a_i} - y_{b_i}|$ independently of the behavior around these values.

To make useful these metrics for time series clustering, (Chouakria-Douzal and Nagabhushan, 2007) propose a dissimilarity index model that include both behavior and observations proximity.

2.2.4 Chouakria-Douzal dissimilarity

(Chouakria-Douzal and Nagabhushan, 2007) introduce a dissimilarity measure addressed to cover both conventional measures for the proximity on observations and temporal correlation for the behavior proximity estimation. The first order temporal correlation coefficient is defined by

$$CORRT(X_t, Y_t) = \frac{\sum_{t=1}^{p-1}(x_{t+1} - x_t)(y_{t+1} - y_t)}{\sqrt{\sum_{t=1}^{p-1}(x_{t+1} - x_t)^2}\sqrt{\sum_{t=1}^{p-1}(y_{t+1} - y_t)^2}},$$

and it is used as a measure of the first order proximity between the dynamic behavior of the series. $CORRT(X_t, Y_t)$ belongs to the interval $[-1, 1]$. The value $CORRT(X_t, Y_t) = 1$ means that both series show a similar dynamic behavior, i.e. their growths (positive or negative) at any instant of time are similar in direction and rate. The value $CORRT(X_t, Y_t) = -1$ implies a similar growth in rate but opposite in direction (opposite behavior). Finally, $CORRT(X_t, Y_t) = 0$ expresses that there is no monotonicity between X and Y, and their growth rate are stochastically linearly independent (different behaviors).

Using this coefficient, they define the following distance:

$$d_{CORRT}(X_t, Y_t) = f[CORRT(X_t, Y_t)]\delta(X_t, Y_t),$$

where $f(\cdot)$ is an adaptive tuning function to modulate a given raw data distance $\delta(X_t, Y_t)$ (like d_{MKW}, d_{FRECH} or d_{DTW}) according to the temporal correlation. The purpose of this dissimilarity measure is weighting the contribution of the similarity of the dynamic behavior of the series and the similarity between their raw values. The conventional raw data discrepancy ($\delta(X_t, Y_t)$) should be increased when the temporal correlation ($CORRT(X_t, Y_t)$) decreases from 0 to -1. The resultant dissimilarity should approach the raw data discrepancy when the temporal correlation is zero. Finally, when the temporal correlation increases from 0 to +1 the raw data discrepancy is decreased. As adaptive tuning function, they choose an exponentially adaptive function given by

$$f(u) = \frac{2}{1 + e^{ku}}, \text{ with } k = 0, 1, 2, \ldots$$

Note that for $k = 0$ the value of d_{CORRT} coincides with the raw distance $\delta(X_t, Y_t)$.

Figure 2.1 shows the effect of the k parameter. As $CORRT(X_t, Y_t)$ tends to 0, $f(u)$ is near 1 for any value of k, making d_{CORRT} approximately equal to $\delta(X_t, Y_t)$. As k increases the contribution of the behavior proximity increases while the contribution the the raw distance decreases.

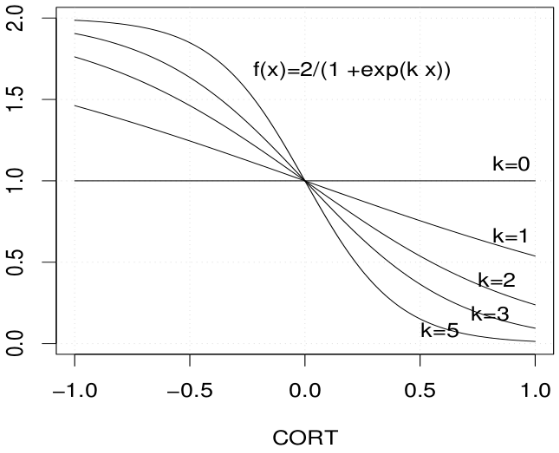


Figure 2.1: Weighting effect based on several values of k.

2.2.5 Autocorrelation-based distance

(Galeano and Peña, 2000) propose a metric based on the estimated autocorrelation function (ACF) for the situations when the correlation structure of the series is of interest.

Let $\hat{\rho_X} = (\hat{\rho}_{1,X}, .., \hat{\rho}_{L,X})^t$ and $\hat{\rho_Y} = (\hat{\rho}_{1,Y}, .., \hat{\rho}_{L,Y})^t$ be the estimated autocorrelation vectors of the time series X and Y, for some L such that $\hat{\rho}_{i,X} \approx 0$ and $\hat{\rho}_{i,Y} \approx 0$ for $i > L$. A distance between time series can be then constructed by means of:

$$d_{ACF} = \{(\hat{\rho}_X - \hat{\rho}_Y)^t \boldsymbol{\Omega} (\hat{\rho}_X - \hat{\rho}_Y)\}^{\frac{1}{2}},$$

where $\boldsymbol{\Omega}$ is a matrix of weights. Some common choices of $\boldsymbol{\Omega}$ are:

- Consider uniform weights by taking $\boldsymbol{\Omega} = \boldsymbol{I}$. In such case, d_{ACF} becomes the Euclidean distance between the estimated autocorrelation functions:

$$d_{ACFU}(X_t, Y_t) = \left\{ \sum_{i=1}^{L} (\hat{\rho}_X - \hat{\rho}_Y)^2 \right\}^{1/2}.$$

- Consider geometric weights decaying with the autocorrelation lag, so that d_{ACF} takes the form:

$$d_{AFCG} = \{\sum_{i=1}^{L} (p(1-p)^i (\hat{\rho}_{i,X} - \hat{\rho}_{i,Y})^2\}^{1/2},$$

  with $0 < p < 1$.

- Consider $\Omega = \mathrm{Cov}(\hat{\rho})^{-1}$, the inverse covariance matrix of the autocorrelations, thus obtaining the Mahalanobis distance between autocorrelations, d_{ACFM}.

Other distances can be introduced by considering the partial autocorrelation functions (PACF's) instead of the ACF's. Hereafter, notation d_{PACFU} and d_{PACFG} will be used to denote the Euclidean distance between the estimated partial autocorrelation coefficients with uniform weights and with geometric weights decaying with the lag (d_{PACFG}), respectively.

All the measures until now work in the time domain, but the frequency domain approach also offers an interesting alternative to measure the dissimilarity between time series. The key idea is to assess the dissimilarity between the corresponding spectral representations of the series.

2.2.6 Periodogram-based distances

(Caiado et al., 2006) introduce several distances based on the periodograms of the time series. Let: $I_X(\lambda_k) = T^{-1}|\sum_{t=1}^{T} X_t e^{-i\lambda_k t}|^2$ and $I_Y(\lambda_k) = T^{-1}|\sum_{t=1}^{T} Y_t e^{-i\lambda_k t}|^2$ be the periodograms of the time series X and Y respectively, at frequencies $\lambda_k = 2\pi k/T$, $k = 1, \ldots, n$, with $n = [(T-1)/2]$. Based on these periodograms several distances are defined below.

The Euclidean distance between the periodogram ordinates:

$$d_P(X_t, Y_t) = \frac{1}{n}\{\sum_{k=1}^{n} (I_X(\lambda_k) - I_Y(\lambda_k)\}^{1/2}.$$

If we are not interested in the process scale, but only on its correlation structure, better results can be obtained using the Euclidean distance between the normalized periodogram ordinates:

$$d_{NP}(X_t, Y_t) = \frac{1}{n}\{\sum_{k=1}^{n}(NI_X(\lambda_k) - NI_Y(\lambda_k)\}^{1/2},$$

where $NI_X(\lambda_k) = I_X(\lambda_k)/\hat{\gamma_0}^X$ and $NI_Y(\lambda_k) = I_Y(\lambda_k)/\hat{\gamma_0}^Y$, with $\hat{\gamma_0}^X$ and $\hat{\gamma_0}^Y$ the sample variance of series X and Y respectively.

Since the variance of periodogram ordinates is proportional to the spectrum value at the corresponding frequencies, it makes sense to use the logarithm of the normalized periodogram:

$$d_{LNP}(X_t, Y_t) = \frac{1}{n}\{\sum_{k=1}^{n}(\log NI_X(\lambda_k) - \log NI_Y(\lambda_k)\}^{1/2}.$$

(Casado de Lucas, 2010) consider a distance measure based on the cumulative versions of the periodograms, i.e. the integrated periodograms. They argue that integrated periodogram based approaches presents several advantages over periodogram based ones, such as:

- Good asymptotic properties. The periodogram is an asymptotically unbiased but inconsistent estimator of the spectral density while the integrated periodogram is a consistent estimator of the spectral distribution.

- From a theoretical point of view, the spectral distribution always exists, the spectral density exists only under absolutely continuous distributions. However, in practice, the integrated spectrum is usually estimated via the estimation of the spectrum.

- The integrated periodogram completely determines the stochastic process.

They propose two distances based on the integrated periodogram, one normalized and other nonnormalized. The normalized version gives more weight to the shape of the curves while the nonnormalized considers the scale. They suggest using the normalized version when the graphs of the functions tend to intersect, and the nonnormalized when they do not. Specifically, the distances based on the integrated periodograms take the form:

$$d_{IP}(X_t, Y_t) = \int_{-\pi}^{\pi} |F_X(\lambda) - F_Y(\lambda)|\, d\lambda,$$

where $F^X(\lambda_j) = C_X^{-1} \sum_{i=1}^{j} I_X(\lambda_i)$ and $F_Y(\lambda_j) = C_Y^{-1} \sum_{i=1}^{j} I_Y(\lambda_i)$, with $C_X = \sum_i I_X(\lambda_i)$ and $C_Y = \sum_i I_Y(\lambda_i)$ for the normalized version, and $C_X = 1$ and $C_Y = 1$ for the nonnormalized version.

2.2.7 Dissimilarity measures based on nonparametric spectral estimators

Also in the frequency domain, (Kakizawa et al., 1998) proposed a general spectral disparity measure given by

$$d_W(X_t, Y_t) = \frac{1}{4\pi} \int_{-\pi}^{\pi} W\left(\frac{f_X(\lambda)}{f_Y(\lambda)}\right) d\lambda,$$

where f_X and f_Y denote the spectral densities of the series X and Y, respectively, and $W(.)$ is a divergence function satisfying appropiate regular conditions to ensure that d_W has the quasi-distance property. If, for example, W is given by:

$$W(x) = \log(\alpha x + (1 - \alpha)) - \alpha \log x$$

with $0 < \alpha < 1$, d_W corresponds to the limiting spectral approximation of the Chernoff information in the time domain. To perform cluster analysis it is necessary to have a symmetrized version of d_W, which can be easily obtained by modifying the divergence function as follows:

$$\tilde{W}(x) = W(x) + W(x^{-1}).$$

In practice, the spectra f_X and f_Y are usually unknown and they must be previously estimated. (Vilar and Pértega, 2004) studied the asymptotic properties of d_W when $f_X(.)$ and $f_Y(.)$ are replaced by non-parametric estimators constructed via local linear regression. These approximations can be done in three different ways ((Fan and Kreutzberger, 1998)) and hence three dissimilarity measures can be also constructed, which are described below.

- $d_{W(DLS)}$, when the spectra are replaced by local lineal smoothers of the periodograms, obtained via least squares.

- $d_{W(LS)}$, when the spectra are replaced by the exponential transformation of local linear smoothers of the log-periodograms, obtained via least squares.

- $d_{W(LK)}$, when the spectra are replaced by the exponential transformation of local linear smoothers of the log-periodograms, now obtained by using the maximum local likelihood criterion. Here, the likelihood function takes the form:

$$L(a,b) = \sum_{-n}^{n} \left[-e^{Y_k - a - b(\lambda_k) - \lambda} + Y_k - a - b(\lambda_k - \lambda) \right] K_h(\lambda_k - \lambda),$$

  where $\lambda_k = \frac{2\pi k}{T}$, $Y_k = \log(I(\lambda_k))$ is the logarithm of the periodogram and K_h is the kernel function with bandwidth h. Since the purpose of this package is automatic clustering, the default value of h is established by the plug-in method for local linear Gaussian kernel regression in (Ruppert et al., 1995).

Another two non-parametric spectral dissimilarity measures studied in (Pértega and Vilar, 2010) are considered. In both cases, the discrepancy measure is given by a non-parametric statistic originally introduced to check the equality of the log-spectra of two processes, that is, to test between

$$H_0 : m_X(.) = m_Y(.)$$
$$H_1 : m_X(.) \neq m_Y(.)$$

with $m_X(\lambda) = \log(f_X(\lambda))$ and $m_Y(\lambda) = \log(f_Y(\lambda))$.

The first distance comes from the generalized likelihood ratio test approach introduced by (Fan and Zhang, 2004) to check whether the density of an observed time series belongs to a parametric family. (Pértega and Vilar, 2010) introduce a slight modification to adjust the procedure to the previously states hypothesis testing to produce:

$$d_{GLK}(X_t, Y_t) = \sum_{k=1}^{n} [Z_k - \hat{\mu}(\lambda_k) - 2\log(1 + e^{\{Z_k - \hat{\mu}(\lambda_k)\}})] - \sum_{k=1}^{n} [Z_k - 2\log(1 + e^{Z_k})],$$

where $Z_k = \log(I_X(\lambda_k)) - \log(I_Y(\lambda_k))$, $\mu(\lambda_k) = m_X(\lambda_k) - m_Y(\lambda_k)$ and $\hat{\mu}(\lambda_k)$ is the local maximum log-likelihood estimator of $\mu(\lambda_k)$ computed by local linear fitting.

The second distance is based on the integrated squared differences between non-parametric estimators of the log-spectra $m_X(\lambda)$ and $m_Y(\lambda)$.

$$d_{ISD} = \int (\hat{m}_X(\lambda) - \hat{m}_Y(\lambda))^2 \, d\lambda,$$

where $\hat{m}_X(\lambda)$ and $\hat{m}_Y(\lambda)$ are the local linear smoothers of the log-periodograms, obtained using the preciously defined maximum local likelihood criterion.

2.3 Model-based approaches

Model-based metrics assume that the underlying models are generated from a particular parametric model. The main approach in literature is to consider that the generating processes follow an invertible ARIMA model. In such case, the clustering procedure usually involves the following steps:

1. fitting an ARIMA model to each time series;
2. measuring the distances between each pair of fitted models;
3. performing cluster based on these distances.

First step requires the estimation of the structure and parameters of ARIMA models. Structure is either assumed to be given or automatically estimated using, for example, Akaike's Information Criterion (AIC) or Schawartz's Bayesian Information Criterion (BIC). Parameters are commonly fitted using the generalized least squares estimators. Some of the most relevant metrics derived in the literature under the assumption of generating models following an ARIMA structure are provided below.

2.3.1 Piccolo distance

(Piccolo, 1990) proposes a metric for the class of invertible ARIMA processes. As an ARIMA model can be correctly specified by an AR(∞) model, Piccolo proposes to measure the discrepancy between two series by means of the Euclidean distance between their corresponding AR(∞) operators. He argues that, in a sense, the π coefficients convey all the useful information about the stochastic structure of the process, since all the other information required is the initial values and the white noise process.

As already mentioned, the AR modeling is automatically performed by using a model selection criterion such as AIC or BIC criterion. Let $\hat{\Pi}_X = (\hat{\pi}_{1,X}, \ldots, \hat{\pi}_{k_1,X})^t$ and $\hat{\Pi}_Y = (\hat{\pi}_{1,Y}, \ldots, \hat{\pi}_{k_2,Y})^t$ be the vectors of AR(k_1) and AR(k_2) parameter estimations of the observed series X_T and Y_T. The Piccolo's distance is calculated as

$$d_{PIC}(X_t, Y_t) = \left\{ \sum_{j=1}^{k} (\hat{\pi}'_{j,X} - \hat{\pi}'_{j,Y})^2 \right\}^{1/2},$$

where $k = \max(k_1, k_2)$, $\hat{\pi}'_{j,X} = \hat{\pi}_{j,X}$, if $j \leq k_1$, and $\hat{\pi}'_{j,X} = 0$ otherwise, and analogously $\hat{\pi}'_{j,Y} = \hat{\pi}_{j,Y}$, if $j \leq k_2$, and $\hat{\pi}'_{j,Y} = 0$ otherwise.

Note that the authors do not consider the residual variance as relevant for comparison, since it is purely a scale parameter, and do not include it in the metric.

2.3.2 Maharaj distance

(Maharaj, 1996) introduced two other discrepancy measures for ARMA processes, based on the hypothesis testing to determine whether or not two time series have significantly different generating processes. More precisely, the hypotheses to be tested are:

H_0: There is no significant difference between the generating processes of two stationary series, i.e $\Pi_X = \Pi_y$.
H_1: There is a significant difference between the generating process of two stationary series, i.e $\Pi_X \neq \Pi_Y$.

The motivation of the test lies in one particular use of the cluster analysis of time series: the identification of a series that is characteristic of all series in a given cluster. Maharaj argues that, since the cluster solution depends on the distance measure, the clustering technique and the analyst, it is useful to have a test to check the homogeneity of the generating models of the series in a given cluster. They further develop this idea and use the p-values of the test to cluster processes that are not significantly different from each other, effectively creating a clustering technique by itself. As consequence of these ideas, both the test statistic and the associated p-value can be used as dissimilarity measures for a more general clustering technique. The proposed test statistic is given by:

$$d_{MAH}(X_t, Y_t) = \sqrt{T}(\hat{\Pi}'_X - \hat{\Pi}'_Y)^t \hat{V}^{-1}(\hat{\Pi}'_X - \hat{\Pi}'_Y),$$

where $\hat{\Pi}'_X = (\hat{\pi}'_{1,X}, \dots, \hat{\pi}'_{k,X})$, $\hat{\Pi}'_Y = (\hat{\pi}'_{1,Y}, \dots, \hat{\pi}'_{k,Y})$, $\hat{V}$ is an estimator of $V = \sigma^2_X R_x^{-1}(k) + \sigma^2_Y R_Y^{-1}(k)$, with σ^2_X and σ^2_Y denoting the variances of the white noise processes associated with X_T and Y_t, and R_X and R_Y are the corresponding sample covariance matrices of both series. k_1 and k_2 are the orders of the AR processes selected by AIC as in d_{PIC}.

(Maharaj, 1996) establishes that, under H_0, $d_{MAH}(X_t, Y_t)$ is asymptotically distributed as a chi-square with k degrees of freedom and, in this way, the values of the statistic can be replaced by the corresponding p-values to construct the metric.

2.3.3 Maharaj extended distance

The second metric proposed by (Maharaj, 2000) does not require the time series are independent and is based on the differences between estimated parameters. Let the

$T-k$ observations of the series X_T and Y_T be expressed by:

$$X = W_x \pi_x + a_x$$

$$Y = W_y \pi_y + a_y$$

where

$x^t = (x_{k+1}, ..., x_{T-1}, x_T)$,

$W_x = \begin{bmatrix} x_k & x_{k-1} & . & . & . & x_1 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ x_{T-2} & x_{T-3} & . & . & . & x_{T-k-1} \\ x_{T-1} & x_{T-2} & . & . & . & x_{T-k} \end{bmatrix}$,

$\pi_x^t = (\pi_{1x}, \pi_{2x}, ..., \pi_{kx})$,

$a_x^t = (a_{k+1x}, ..., a_{T-1x}, a_{Tx})$.

The quantities y^t, W_y, π_y^t and a_y^t are analogously defined. Furthermore, $E[a_x] = 0$, $E[a_x a_x^t] = \sigma_x I_{T-k}$, $E[a_y] = 0$ and $E[a_y a_y^t] = \sigma_y I_{T-k}$. The measure assumes that the disturbances of the two models are correlated at the same points in time but with uncorrelated across observations. That is:

$$E(a_x a_y^t) = \sigma_{xy} I_{T-k}.$$

Assuming a total of $2(T-k)$ observations are used, the combined model may be expressed as:

$$Z = W\pi + a,$$

where $Z = \begin{bmatrix} X \\ Y \end{bmatrix}$, $W = \begin{bmatrix} W_x & 0 \\ 0 & W_y \end{bmatrix}$, $\pi = \begin{bmatrix} \pi_x \\ \pi_y \end{bmatrix}$ and $a = \begin{bmatrix} a_x \\ a_y \end{bmatrix}$. Now, it is observed that $E(a) = 0$ and $E(aa^t) = V = \Sigma \otimes I_{T-k}$, with $\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}$.

The proposed test statistic is:

$$D = (R\hat{\pi})^t [R(W^t \hat{V}^{-1} W) R^t]^{-1} (R\hat{\pi}),$$

with $R = [I_k, -I_k]$, $\hat{V} = \hat{\Sigma} \otimes I_{T-k}$, $\hat{\pi} = [W^t \hat{V}^{-1} W]^{-1} W^t \hat{V}^{-1} Z$.

As $D \sim \chi_k^2$, the p-value computed on D is the final distance metric:

$$d_{MAHEXT}(X_y, Y_t) = P(D \geq \chi_k^2).$$

2.3.4 Cepstral-based distance

(Kalpakis et al., 2001) propose a dissimilarity measure based on the Linear Predictive Coding (LPC) cepstrum. The cepstrum is defined as the inverse Fourier transform of the short-time logarithmic amplitude spectrum. The cepstrum defined using the autoregression coefficients from linear model of the signal is referred to as the LPC Cepstrum, since it is derived from the linear predictive coding of the signal.

Consider a time serie X_t defined by an $AR(p)$ model:

$$X_t = \sum_{r=1}^{p} \phi_r X_{t-t} + \epsilon_t,$$

where ϕ_r are the autoregression coefficients and ϵ_t is white noise with 0 mean and non-zero variance. The LPC cepstral coefficients are defined by:

$$\psi_h = \begin{cases} \phi_1 & : h = 1 \\ \phi_h + \sum_{m=1}^{h-1}(\phi_m - \psi_{h-m}) & : 1 < h \leq p \\ \sum_{m=1}^{p}(1 - \frac{m}{h})\phi_m \psi_{h-m} & : p < h \end{cases}$$

The distance will then be the Euclidean distance between cepstral coefficients.

$$d_{CEP} = \{\sum_{i=1}^{T}(\psi_i^X - \psi_i^Y)^2\}^{1/2},$$

with ψ^X and ψ^Y the cepstral coefficients of the series X_t and Y_t respectively.

2.4 Prediction-based approaches

(Alonso et al., 2006) and (Vilar et al., 2010) do not consider an approach based on the raw data or features of the series neither just on the model that generates it, but in their forecast at an specific point in time. Figure 2.2 shows a scenario where clustering based on: (i) the underlying models, (ii) the last observed values or (iii) the forecasts at an specific horizon, produce totally different results. The prediction-based approach is specifically suited to clustering scenarios where the interest is in the long term convergence or where some specific level is going to be reached. (Alonso et al., 2006) use the CO_2 emission reduction of the Kyoto Protocol to illustrate the applicability of this method, since in this particular case a method based on the properties of the predictions is clearly interesting. The idea of using series forecast introduces an extra consideration on the clustering problem, namely which point in the future will be considered. The introduction of this parameter makes this approach notably different to the other distance metrics in this work. In fact, the value of this parameter is selected by the user from the context but externally to the information conveyed by the series, while the other metrics only work with the provided series.

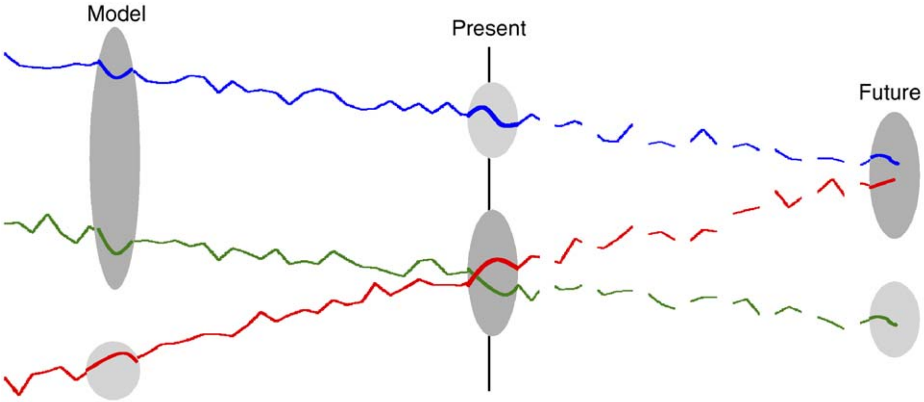


Figure 2.2: Three different cluster solutions depending on whether modelling, present information or future values are used.

This clustering procedure is based on the full forecast densities instead of focusing on the pointwise forecast. The differences between each pair of densities are used to fill a dissimilarity matrix from which the cluster analysis will be carried out. One of the advantages of considering the full forecast densities is the ability to distinguish between series generated by models that are essentially similar, e.g. models differing only in the variability of the observations or in the distribution of the innovations, but which produce different forecast densities. For instance, forecast densities on left panel in Figure 2.3 have equal means but showing a lower similarity index than the ones on right panel, where the means are different. Note also that for constructing the forecasts, both present and past information are used so no valuable knowledge is discarded.

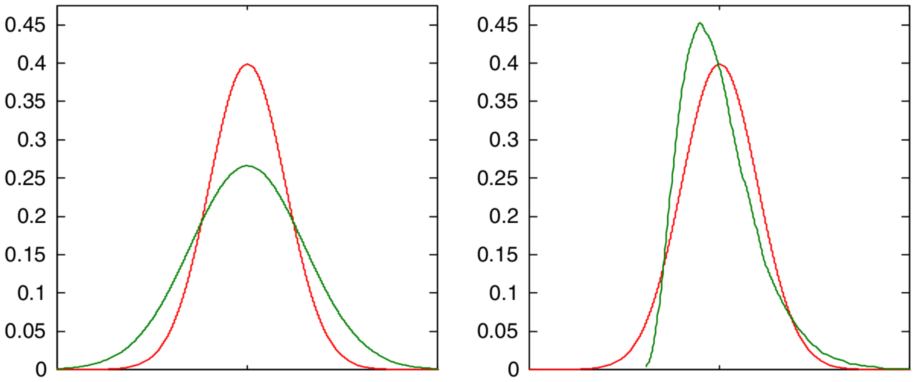


Figure 2.3: Considering only point forecast can produce incorrect results.

In this work, the most general approach proposed by (Vilar et al., 2010) is considered. Their approach is valid to be applied to general autoregressive models, including extensively studied parametric models, such as the threshold autoregressive (TAR), the exponential autoregressive (EXPAR), the smooth-transition autoregressive (STAR) and the bilinear, among others, see (Tong and Yeung, 2000) and references therein.

2.4.1 Distance procedure

Let Ξ be a class of real value stationary processes $\{X_t\}_{t\in\mathbb{Z}}$ such that

$$X_t = m(X_{t-1}) + \epsilon_t$$

where

- $\{\epsilon_t\}$ is and i.i.d. sequence.
- X_{t-1} is a d-dimensional vector of known lagged values.
- $m(.)$ is assumed to be a smooth function, not restricted to any pre-specified parametric model.

We wish to perform a cluster analysis on a set S of s partial realizations from series belonging to Ξ, i.e. each series in study is generated from a process satisfying the above model. The goal of the clustering process is to capture the similarities in the forecasts at a specific future time $T+b$. Given two series X and Y in S, the distance between them is defined by:

$$D_{L^1}(X_t, Y_t) = \int |f_{X_{T+b}}(u) - f_{Y_{T+b}}(u)|\, du,$$

where $f_{X_{T+b}}$ and $f_{Y_{T+b}}$ denote the densities of the forecasts X_{T+b} and Y_{T+b}, respectively.

The L^1 functional distance is chosen over the L^2 because the last removes the effect of the distance between the point forecasts and is only governed by the shape of the forecast densities. It is not useful when the forecast densities at the specified horizon are disjoint. On the othe hand, when this happens, $D_{L^1} = 2$, allowing the production of reasonable clusters.

Since direct computation of the distance is not feasible in practice due to the unknown nature of the forecast densities, the densities are replaced by the kernel-type estimators based on bootstrap predictions. A detailed description of the steps involved in generating the bootstrap predictions is given below.

1. Estimate m using a Nadaraya-Watson estimator $\hat{m}_{g_1}$.
2. Compute the nonparametric residuals, $\hat{\epsilon}_t = X_t - \hat{m}_{g_1}(X_{t-1})$
3. Construct a kernel estimate, $\hat{f}_{\tilde{\epsilon},h}$ of the density function associated to the centered residuals $\tilde{\epsilon} = \hat{\epsilon}_t - \hat{\epsilon}_\bullet$
4. Draw a bootstrap resample ϵ_t^* of i.i.d from $\hat{f}_{\tilde{\epsilon},h}$.
5. Define the bootstrap series X_t^* by the recursion $X_t^* = \hat{m}_{g_1}(X_{t-1}^*) + \epsilon_t^*$
6. Obtain the bootstrap autoregressive function $\hat{m}_{g_2}$, using the bootstrap sample $(X_1^*, ..., X_T^*)$
7. Compute bootstrap prediction paths by the recursion $X_t^* = \hat{m}_{g_2}(X_{t-1}^*) + \epsilon_t^*$ for $t = T+1, .., T+b$ and $X_t^* = X_t$ for $t \leq T$
8. Repeat steps 4-7 for the desired amount B of bootstrap resamples.

Applying the resampling method to X, provides a bootstrap sample $(X_{T+b}^{*1}, ..., X_{T+b}^{*B})$ from wich the unknown density of X_{T+b} can be estimated using kernel techniques. In particular, the Rosenblatt-Parzen kernel smoother is used to obtain $\hat{f}_{X_{T+b}^{(i)*}}(x)$, the b-step ahead estimator at point x of X. Finally the distance D_{L^1} can be approximated by the plug-in version:

$$d_{PRED} = \hat{D}_{1,ij}^* = \int |\hat{f}_{X_{T+b}^{(i)*}}(x) - \hat{f}_{X_{T+b}^{(j)*}}|dx$$

This procedure is valid for series of unequal length, unlike other dissimilarity measures.

Chapter 3

Clustering

There are a lot of different taxonomies of clustering algorithms (Berkhin, 2006), (Everitt et al., 2001), (Hansen and Jaumard, 1997), (Jain and Dubes, 1988) but most authors coincide (Xu et al., 2005), (Kalpakis et al., 2001),(Everitt et al., 2001), (Jain et al., 1999) in distinguishing two great families of clustering algorithms: partitioning and hierarchical.

We will give a simple mathematical definition of these two types.

Given a set of input patterns $X = \{x_1, x_2, ..., x_N\}$ where $x_j = (x_{j1}, x_{j2}, ..x_{jd})^T \in \mathbb{R}$ and each measure x_{jd} is said to be a feature.

- **Partitioning** cluster attemps to seek a K-partition of X, $C = \{C_1, C_2, ..C_K\}$, $K \leq N$ such that
  1. $C_{i_K} \neq \emptyset$, $i = 1, ..., K$
  2. $\cup_{i=1}^{K} C_i = X$
  3. $C_i \cap C_j = \emptyset, i, j = 1, ...K, i \neq k$
- **Hierarchical** clustering attemps to construct a tree-like nested partition of X, $H = \{H_1, ..., H_Q\}$, $Q \leq N$, such that $C_i \in H_m$, $C_j \in H_l$ and $m > l$ imply $C_i \in C_j$ or $C_i \cap C_k = \emptyset$ for all $i, j \neq i, m, l = 1, ...Q$.

Among partitioning cluster algorithms, K-means and K-medoids are the two most representative, and for the hierarchical clustering algorithm, agglomerative with single-linkage or complete-linkage (Xu et al., 2005).

Most common clustering algorithms, including those mentioned above, are general-purpose and so can be applied to time series clustering using the appropiate distance measure for computing the dissimilarity matrix. Implementations exist in **R** packages such as (R Development Core Team) and (Maechler et al., 2011). These implementations are constantly being tested by the whole **R** community and have acquired a degree of maturity that makes it counter productive to reproduce their work. Nevertheless, we have included in our work a clustering algorithm, explained in the following section, that was created in the context of time series clustering.

3.1 Maharaj Clustering Algorithm

This clustering algorithm was introduced by (Maharaj, 2000) for the use with their proposed statistic for the hypothesis on series coming from the same model. The first step is performing the test on each pair of series and determine the p-value. Then the series are grouped together in way similar to agglomerative hierarchical clustering. The main differences lie series are only grouped together if their p-values exceed a previously stablished significance level. Serie y can only join the cluster of serie x if the p-value of the pair (x, y) and every other combination of the elements in the cluster of x with y exceeds the significance level. In the same way, two clusters are only merged if every combination of elements across the two clusters are greater than the significance level. The full description of the algorithm can be seen in Figure 3.1. This clustering procedure can be applied to any dissimilarity metric based on p-values.

3.2 Clustering results evaluation criteria

Altough it can be argued that there is no true correct or incorrect clustering solution, it is helpful to have some criteria to evaluate different clustering methods. Two different criteria can be distinguished: known ground-truth and unknown ground-truth. The number of desired clusters and the classification of each element for a particular problem must be known for the former method, and no extra information is required for the latter. The most common method for each type of criteria are explained in the following sections.

3.2.1 Known Ground-truth cluster evaluation criteria

3.2.1.1 Gavrilov similarity

For this kind of criteria one metric is selected, Gavrilov similarity (Gavrilov et al., 2000). It is computed as follows:

Let $G = G_1, ..., G_K$ be the "ground-truth" cluster and $A = A_1, ..., A_k$ be the cluster obtained using some clustering method that we want to test, then cluster similarity metric $Sim(G, A)$ is:

$$Sim(G, A) = (\sum_i (\max_j Sim(G_i, A_j)))/k$$

with:

$$Sim(G_i, A_j) = \frac{|G_i \cap A_j|}{|G_i| + |A_j|}$$

3.2.1.2 Rand index

The Rand index (Rand, 1971) is a measure of the similarity between two clusterings. Let S be a set of N data elements. Given two clustering of S, namely $A = \{A_1, ..., A_R\}$ with R clusters and $B = \{B_1, ..., B_C\}$ with C clusters, the Rand index is computed over the pairs of elements on wich the two clustering agree or disagree. Any pair of data elements of S of the total of $\binom{N}{2}$ distinc pairs falls into one of these four categories :

1. N_{11}: the number of pairs that are in the same cluster in both A and B
2. N_{00}: the number of pairs that are in different clusters in both A and B.
3. N_{01}: the number of pairs that are in the same cluster in A but in different clusters in B.
4. N_{10}: the number of pairs that are in different clusters in A but in the same cluster in B.

The Rand index is defined by:

$$RI(A, B) = \frac{N_{00} + N_{11}}{\binom{N}{2}}$$

The Rand index lies between 0 and 1. It takes the value 1 when the two clusters are identical and 0 when no pair of points appear in the same cluster.

3.2.1.3 Adjusted Rand index

(Hubert and Arabie, 1985) propose an adjusted-by-chance modification of the Rand index by taking the hypergeometric distribution as the model of randomness. The information on cluster overlap can be summarized in the form of a $R \times C$ contingency table as in table 3.1.

A/B	B_1	B_2	$\cdots$	B_C	Sums
U_1	n_{11}	n_{12}	$\cdots$	n_1C	a_1
U_2	n_{21}	n_{22}	$\cdots$	n_2C	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
U_R	n_{R1}	n_{R2}	$\cdots$	n_RC	a_R
Sums	b_1	b_2	$\cdots$	b_C	$\sum_{ij} n_{ij} = N$

Table 3.1: Contingency table for pair overlap information.

The adjusted Rand index is then defined by:

$$AdjustedIndex = \frac{Index - ExpectedIndex}{MaxIndex - ExpectedIndex}$$

more specifically:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{N}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{N}{2}}$$

The adjusted Rand index lies between -1 and 1, taking the value 0 when the index takes the expected value.

3.2.2 Unknown Ground-truth cluster evaluation criteria

3.2.2.1 Silhouette width

Silhoutte width, first described by (Rousseeuw, 1987) does not require a true reference solution, it is only a measure on the degree of uniformity of the clusters. It is defined as follows. Let A be the cluster to which the object i belongs. Let $a(i)$ be the average dissimilarity measure of i to all the the objects of A. Consider any cluster C different from A. Let $d(i, C)$ be the average dissimilary measure of i to all the objects of C. Take the smallest of those average dissimilarity $b(i) = \min_{C \neq A} d(i)$. The cluster B which attains this minimum $d(i, B) = b(i)$ is called neighbor of object i, the second best cluster for object i. The value $s(i)$ is defined by:

$$s(i) = \frac{a(i) - b(i)}{\max(a(i), b(i))}$$

The value $s(i)$ lies between 1 and -1. Values close to 1 mean the object is well clustered, and if it is closer to -1 the object is badly clustered.

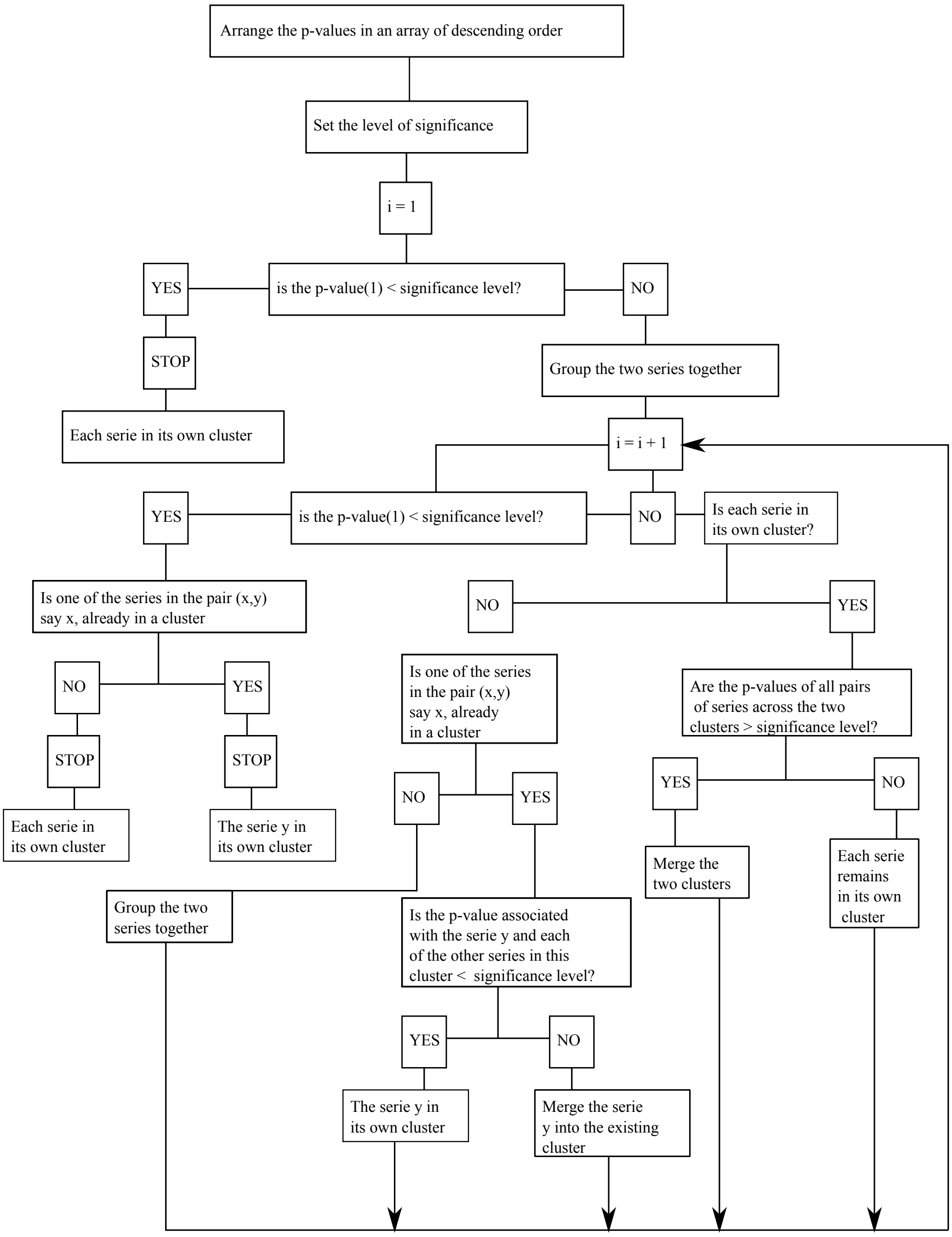

Arrange the p-values in an array of descending order
Set the level of significance
i = 1
YES
is the p-value(1) < significance level?
NO
STOP
Group the two series together
Each serie in its own cluster
i = i + 1
YES
is the p-value(1) < significance level?
NO
Is each serie in its own cluster?
Is one of the series in the pair (x,y) say x, already in a cluster
NO
YES
NO
YES
Is one of the series in the pair (x,y) say x, already in a cluster
Are the p-values of all pairs of series across the two clusters > significance level?
STOP
STOP
NO
YES
YES
NO
Each serie in its own cluster
The serie y in its own cluster
Merge the two clusters
Each serie remains in its own cluster
Group the two series together
Is the p-value associated with the serie y and each of the other series in this cluster < significance level?
YES
NO
The serie y in its own cluster
Merge the serie y into the existing cluster

Figure 3.1: Maharaj clustering algorithm for time series.

Chapter 4

Simulation study

The purpose of the simulation study is threefold. First, it compares with the results shown in existing research papers such as (Pértega and Vilar, 2010), checking for the correctness of the implementation. Second, it studies the performance measures that have not been compared with other proposals such as (Casado de Lucas, 2010). Finally it gives a general reference for the performance of the measures.

4.1 Classification of time series as Stationary or Non-stationary

The purpose of this experiment, performed in (Caiado et al., 2006) and extended in (Pértega and Vilar, 2010) for a subset of the dissimilary measures targeted in this work, is testing the performance of a given measure in classifying series as stationary or non-stationary.

The procedure is as follows: Given a set of time series, create two clusters using the studied measure. Each of the time series to classify has been simulated from a chosen known model, of which the stationary property is known. For each of the measures to study, compute the dissimilary matrix of the series generated from their models and cluster them in two groups using complete linkage agglomerative hierarchical clustering. The resulting group with greater amount of series that come from stationary models is considered the stationary group, and the other the non-stationary group. The final result is the ratio of series that have been grouped in the right cluster. The reason behind using this kind of hierarchical clustering and not other lies in the lack of proper interpretation for the methods that use average of elements in a group and similar approaches. Taking the Piccolo distance as an example, an representative element of a cluster formed by averaging their existing elements has no real interpretation in terms of similarity with another given outside element.

The series used in this study come from a general ARIMA(p, d, q) model defined by:

$$\phi(B)(1-B)^d M_t = \theta(B)\omega_t, t = 0, \pm 1, ...$$

where B is the back-shift operator such that $B^t M_t = M_{t-r}$, $\phi(B) = 1 - \phi_1 B - ...\phi_p B^p$ is the p-order autoregressive operator, $\theta(B) = 1 - \theta_1 B - ...\theta_q B^q$ is the q-order moving average operator, d is the order of differenciating (so that $d = 0$ for a stationary process, $d \geq 1$ for a non-stationary process) and ω_t is a sequence of independent variables with constant mean and variance. Each process $N_t = (1-B)^d M_t$ is assumed to be causal and invertible. As in (Pértega and Vilar, 2010) and (Caiado et al., 2006), one realization of the following 12 ARIMA processes, six stationary and six non-stationary, was generated.

(a) $AR(1)$ $\phi_1 = 0.9$

(b) $AR(2)$ $\phi_1 = 0.95, \phi_2 = -0.1$

(c) $ARMA(1,1)$ $\phi_1 = 0.95, \theta_1 = 0.1$

(d) $ARMA(1,1)$ $\phi_1 = -0.1, \theta_1 = -0.95$

(e) $MA(1)$ $\theta_1 = -0.9$

(f) $MA(2)$ $\theta_1 = -0.95, \theta_2 = -0.1$

(g) $ARIMA(1,1,0)$ $\phi_1 = -0.1$

(h) $ARIMA(0,1,0)$

(i) $ARIMA(0,1,1)$ $\theta_1 = 0.1$

(j) $ARIMA(0,1,1)$ $\theta_1 = -0.1$

(k) $ARIMA(1,1,1)$ $\phi_1 = 0.1, \theta_1 = -0.1$

(l) $ARIMA(1,1,1)$ $\phi_1 = 0.05, \theta_1 = -0.05$

In all cases, the error was Gaussian white noise with zero mean and unit variance. The experiment was performed on series sampled from the models above, for three different lengths $T = (50, 200, 500)$. For each length, the experiment was repeated $N = 300$ times. Table 4.1 shows the average success percent of the clustered series over the T repetitions. Since we focus on automatic clustering, all the distances that require extra parameters, such as the decaying rate of ACFG distance, bandwiths, . . . select them in a unassisted way. The results show a general tendency to improve the classification with the growth of the sample size T, in particular autocorrelation based metrics and LNP show the stronger benefits. No measure decreases its performace with the increase of T. The success rate of 0.75 is very common among the studies metrics, the reason beign that those metric find a strong dissimilarities among the sampled series that do not change across repetitions of the experiment. Paricularly, the first two models are the ones that produce this phenomenon by beign constantly grouped in the nonstationaty cluster, with one of the two $ARMA$ models causing the third erroneus classification to achieve the 0.75 success rate. The best performing distance is ACFG with a success rate of 0.93 for T, but it is worth mentioning that for smaller sample sizes $T = 50$, it performs similar to the 0.75 success rate of other metrics, and it could perform even worse for small series . It is a noteworthy observation that some of the measures are not affected by this and so are preferred if that is the case. The worse performing distance is the one based on cepstral coefficients, it is highly dependant on the automatic model selected. Among the Chouakria temporal correlation coefficient distances, Dynamic Time Warping gets betters results than Frechet.

The results are consistent with those shown in (Pértega and Vilar, 2010) and the differences fade after selecting the same parameters that they used in their simulation.

	T		
Measure	50	200	500
Model-free			
d_{ACFU}	0.79	0.89	0.91
d_{ACFG}	0.8	0.87	0.93
d_{PACFU}	0.74	0.75	0.75
d_{PACFG}	0.75	0.75	0.75
d_P	0.6	0.64	0.66
d_{LP}	0.67	0.68	0.69
d_{NP}	0.62	0.64	0.67
d_{LNP}	0.77	0.87	0.91
$d_{W(DLS)}$	0.75	0.75	0.75
$d_{W(LK)}$	0.75	0.75	0.75
d_{ISD}	0.75	0.75	0.75
d_{GLK}	0.75	0.75	0.75
$d_{Integrated}$	0.76	0.75	0.75
$d_{CHOUAK.FRECH}$	0.66	0.66	0.66
$d_{CHOUAK.DTW}$	0.67	0.69	0.69
Model-based			
d_{PIC}	0.75	0.75	0.75
d_{MAH}	0.75	0.75	0.75
d_{MAHEXT}	0.64	0.66	0.70
d_{CEPST}	0.58	0.58	0.58

Table 4.1: Results of the stationary or non-stationary classification.

4.2 Classification of Non-linear time series

This experiment consists in the classification of non-linear time series, to study how the metrics work in this scenario. These series were simulated from four different underlying processes:

- Model(1): Threshold Autoregressive (TAR) model:

$$X_t = 0.5X_{t-1}I(X_{t-1} \leq 0) - 2X_{t-1}I(X_{t-1} > 0) + \epsilon_t$$

- Model(2): Exponential Autoregressive (EXPAR) model:

$$X_t = (0.3 - 10e^{-X_{t-1}^2})X_{t-1} + \epsilon_t$$

- Model(3): Linear Moving Average (MA) model:

$$X_t = \epsilon_t - 0.4\epsilon_{t-1}$$

- Model(4): Non-linear Moving Average (NLMA) model:

$$X_t = \epsilon_t - 0.5\epsilon_{t-1} + 0.8\epsilon_{t-1}^2$$

With ϵ_t a process consisting in independent zero mean and unit variance Gaussian variables. These models are used in (Tong and Yeung, 2000) for linearity tests. Note that Model(3) is a linear process. The clustering objective is grouping series with the same underlying models in a set of sampled series from all the four given models. Since the true solution of the cluster is know, the Gravrilov index is used. The clusters were created using the hierarchical clustering algorithm with complete linkage until four clusters are left. The experiment was repeated 100 times for each distance metric, with the results shown in Table 4.2 being the average Gravrilov index obtained. The adjusted Rand index is also included. Cepstral distance works noticeably better in this context, compared with the stationarity test. The rest of the model-based metrics suffer from the lack of linearity, since they are also based on AR models. While they have close Gravrilov indices, the adjusted Rand index show that they have inferior performance to the Cepstral distance. Autocorrelation and Partial-Autocorrelation based distance have average performance, attributed to the non-linearity of the processes. The reason behind the low performance of the periodogram based distances is their sensibility to the high variance that affect the periodogram . This effect is mitigated by the spectral smoothers, and, as shown in (Pértega and Vilar, 2010), non-parametric metrics exhibit superior performance in this context. The adjusted Rand index has been proved useful for finer grain comparison of measures with close Gravilov indices.

Measure	Gavrilov index	Adj. Rand index
Model-free		
d_{ACFU}	0.67	0.18
d_{ACFG}	0.70	0.26
d_{PACFU}	0.68	0.16
d_{PACFG}	0.65	0.19
d_P	0.62	0.10
d_{LP}	0.61	0.1
d_{NP}	0.64	0.12
d_{LNP}	0.66	0.1
$d_{W(LS)}$	0.83	0.55
$d_{W(LK)}$	0.87	0.63
d_{ISD}	0.83	0.62
d_{GLK}	0.81	0.55
$d_{Integrated}$	0.75	0.34
$d_{CHOUAK.FRECH}$	0.72	0.31
$d_{CHOUAK.DTW}$	0.75	0.42
Model-based		
d_{PIC}	0.64	0
d_{MAH}	0.64	0
d_{MAHEXT}	0.64	0
d_{CEPST}	0.7	0.23

Table 4.2: Results of the non-linear classification.

Chapter 5

Clustering Real Datasets

While a simulation study is better suited for a comparison between metrics, real datasets also offer a good benchmarking opportunity. There is also the need to illustrate the performance of the forecast-based distance included in this work. Since comparing this last group of distances with those on the model-free and model-based metrics is not appropiate, two examples have been selected for each group. The particular datasets are chosen from the works in (Kalpakis et al., 2001) for the model-free and model-based metrics, and (Alonso et al., 2006) and (Vilar et al., 2010) for the forecast-based one. All these datasets have in common that they require some kind of transformation before applying clustering procedure. Even though, arguably, not all of the implemented dissimilarity measures require transformations, we follow the same steps suggested in these works for a better comparison.

Kalpakis et al. (Kalpakis et al., 2001) work with several real datasets, and use two different methods to evaluate their metrics. The first method is based on a similarity metric of the generated clusters when compared with a given "ground-truth" or "true" cluster, the Gavrilov index. The second measure is the average silhouette width, a notion of the stability of a particular clustering solution.

Each measure is used to compute a dissimilary matrix, that is the base of a partitioning against medoids (PAM) (Maechler et al., 2011) clustering method. We have used hierarchical clustering with complete linkage due to lack of interpretation of the centroid of a group.

5.1 Population Dataset

This dataset is composed of 20 time series that represent the estimate population from 1900 to 1999 in 20 states of the US. The authors distinguish two main clusters, the first is composed by states that have exponentially increasing trend and the second is composed by states with a stabilizing trend. A representative example of elements of both clusters is shown in Figure 5.2.

This classification sets the true cluster that will be used for the "ground-truth" cluster in the gravrilov similarity metric. The series of the dataset are non-stationary, and are subject by the authors to a set of transformation to make them stationary,

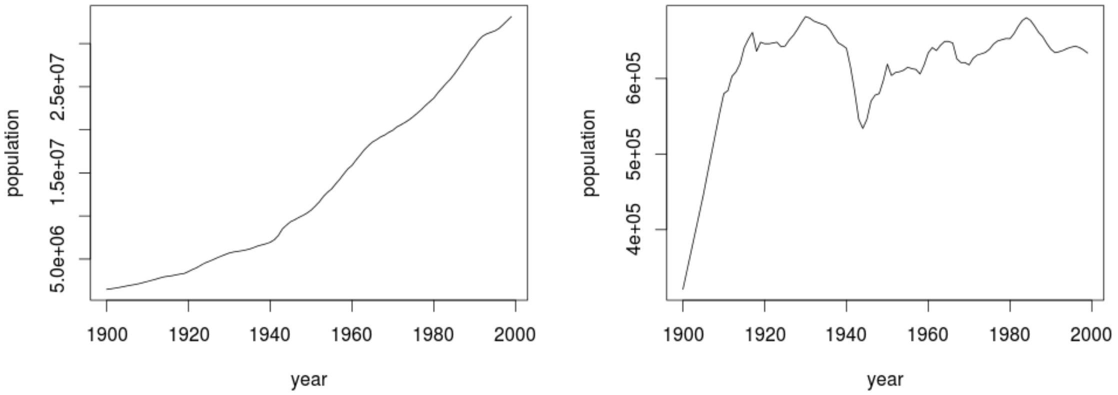


Figure 5.1: Example of time series for a state in the exponential (California) cluster and a state in the stabilizing cluster (North Dakota).

a requisite of some of the time series dissimilarity measures that are studied. The transformation is done according with the following steps:

1. The series are smoothed by using a window average over a window size of 2. This helps in reducing the high frequency noise.

2. The non-stationarity over the variance is lessened by taking a logarithmic transform of x_{avg}, $\log x_{avg}$, with x_{avg} the smoothed serie coming from the previous step.

3. After these two transforms, there is still a non-stationary mean that suggest a differenciating step. The result is $x_{diff} = (1 - B) \log x_{avg}$ where B is the back-shift operator.

The Table 5.1 shows the result of the clustering for a representative set of metrics. A illustrative output of clustering for the metric $ACFG$ is shown in Figure 5.2. The top three metrics are the logarithm of normalized periodogram, autocorrelation with geometrically decaying weigths and the integrated periodogram ones. Despite that LNP produces the best solution in terms of Gavrilov index, the low average silhouette shows high inestability and with other but similar data, its result could be quite different. Integrated periodogram could be a more reliable method in this context. The overall success rate of the metrics of 0.7 evidences a difficult scenario, that could be atributed to the general nature of the transformations applied to the data. Model-based approaches suffer once again from the missespecification of their parameters while showing good silhoutte coefficients compared with the model-free approaches.

Measure	$Sim(G, A)$	$SilhouetteWidth$
Model-free		
d_{ACFG}	0.74	0.79
d_{PACFU}	0.65	0.67
d_{LNP}	0.85	0.28
$d_{INTEG.PERIOD}$	0.8	0.53
$d_{W.LK}$	0.65	0.65
$d_{CHOUAK.DTW}$	0.62	0.52
Model-based		
d_{MAH}	0.66	0.88
$d_{CEPSTRAL}$	0.66	0.76

Table 5.1: Results of the Population dataset clustering.

5.2 Electrocardiogram Dataset

This dataset features three different groups of EGC time series obtained from PhysioNet (Goldberger et al., 2000 (June 13). The electrocardiograms are taken at intervals of 8 millisecond over a period of two seconds. The first group is composed by recordings of people having malignant ventricular arrythmia. The arrythmia is a change in the normal rythm of the heart. This kind of arrythmia is the most serious that exists since it is equivalent to a cardial arrest. The second group includes recordings of healthy people. The third group is formed by recordings of people having supraventricular arrythmia. This kind of arrythmia is usually benign. Figure 5.3 shows one element of each group. As is clearly seen in the Figure, the data is nonstationary. (Kalpakis et al., 2001) work with this dataset and apply a series of three consecutive differenciations to obtain stationarity. For the test, we take 10 samples from each of the previously described groups. The objective of the clustering will be group the elements according to their original groups: ventricular, healthy and supraventricular. Since the correct solution is known beforehand, a ground-truth based performance metric will be used. The clustering algorithm selected is agglomerative hierarchical clustering with complete linkage, taking the last three remaining clusters as the solution. The results of the test are shown in Table 5.2 and a representative solution in Figure 5.4. For this particular problem, $ACFG$ is clearly the best metric, having a superior Gravilov index while also obtaining a high silhouette coefficient. Cepstral based distance achieves a good silhouette coefficient, but a poor success ratio. It is observed that series are well-separated but they are not correctly grouped according to the needs of this context. A particular characteristic of this problem is that healthy readings tend to get clustered together, often creating a perfect cluster, and that the majority of the mistakes come from locating some arrythmia samples into the wrong group. LNP metric exhibits an extremelly low silhouette coefficient due to some of their elements having negative values. Bening arrythmia

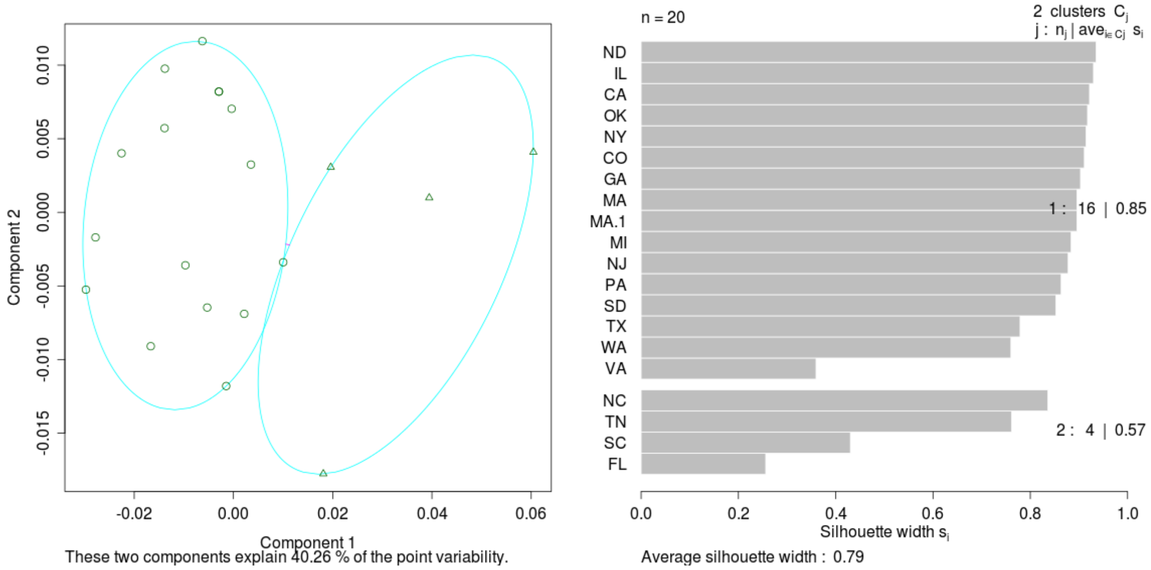
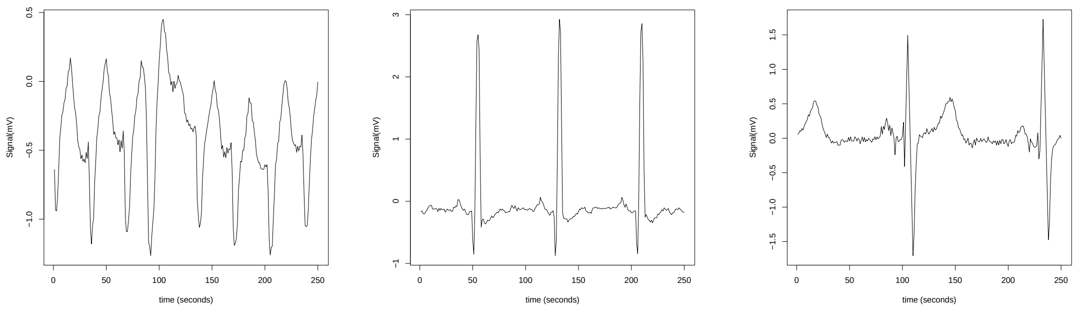


Figure 5.2: Result of clustering real data population series using the d_{ACFG} distance measure.

recordings have also their first half very similar to their healthy counterparts, and this fact could be behind the relatively low silhouette coefficient of the metrics.

5.3 Forecast

Due to the unique nature of the forecast-based distances, it is not appropiate to compare them with dissimilarity metrics that follow different viewpoints. The datasets



(a) Ventricular arrythmia (malign).

(b) Healthy electrocardiogram

(c) Supraventricular arrythmia (benign).

Figure 5.3: Samples of ECG readings

Measure	$Sim(G, A)$	$SilhouetteWidth$
Model-free		
d_{ACFU}	0.6	0.27
d_{ACFG}	0.73	0.51
d_{PACFU}	0.7	0.22
d_{LNP}	0.6	0.07
$d_{INTEG.PERIOD}$	0.6	0.5
$d_{CHOUAK.DTW}$	0.5	0.3
$d_{W.LK}$	0.65	0.48
Model-based		
d_{PIC}	0.52	0.52
d_{MAH}	0.6	0.35
$d_{CEPSTRAL}$	0.52	0.9

Table 5.2: Results of the ECG dataset clustering.

should also portrait a scenario with interesting possibilites for a cluster based on predictions. We have chosen two datasets, studied in (Alonso et al., 2006) and (Vilar et al., 2010), as a meaning to show the performance of this type of distance metric. The first dataset consists in annual $C0_2$ emissions taken between 1960 and 1999 for countries facing the Kyoto agreement. The second dataset is formed by mothly industrial production indices by countries memeber of the Organization for Economic Cooperation and Development (OECD).

5.3.1 CO_2 emissions dataset

The Kyoto agreement dictates a reduction in emission for the participating countries by the year 2012, making natural to set the forecast horizon at this point in time. 2012 is also 12 years from the ending datum of each series, presumably enough time to deviate from this last point and produce interesting results from the prediction point of view.

The emissions can be seen in Figure 5.5. Some countries like China or Korea feature growing patterns, others as USA exhibit a relatively stable behavior and others are clearly decreasing, with Great Britain and France among them.

Due to the non-stationarity of the data, a chain of transformations is required in order to produce stationarity, a requisite of the bootstrap method used in the metric. The transformations are obtained from TRAMO (Time series Regression with ARIMA noise, Missing observations and Outliers) (Gómez and Maravall, 1996), a program developed to automatically identify log/level transformations and the presence of calendar-type effects, as well as detecting and correcting additive outliers, transitory changes and level shifts in underlying ARIMA models. Once the transformations are properly identified, the procedure for applying the distance measure to

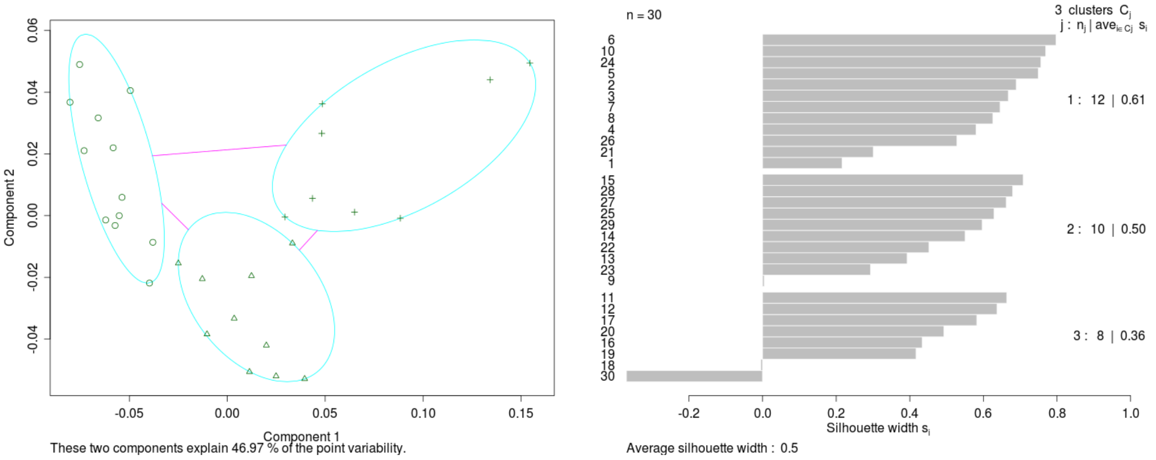


Figure 5.4: Result of clustering the ECG series using the d_{ACFG} distance measure.

the dataset is as follows:

1. Apply the identified transformation to the data.
2. Use the bootstrap method to generate predictions resamples at the given horizon (year 2012).
3. Backtrasform the bootstrap predictions.
4. Calculate the densities of the predictions.
5. Compute the L_1 distance for each pair of densities calculated from the back-transformed bootstrap predictions.

Once the dissimilarity matrix is computed, a hierarchical clustering with average linkage procedure is applied, producing the results shown in Figure 5.6. In Figure 5.7 the estimation of the forecast densities is depicted. If we take the last available observation as a reference, we can see that countries like Australia and Canada swap their positions as the second and third top emitter. Israel and Venezuela, with almost identical point forecasts are not grouped together first beacuse the density of Israel is more similar to France's. Argentina and Mexico are grouped together before Argentina and Korea, despite these last two having closer point forecasts. Japan and Saudi Arabia feature similar behavior, with high variance, and are grouped together at an early stage, but much later to the rest of the countries with a similar point forecasts.

5.3.2 Industrial production indices dataset

The second dataset consists in monthly industrial production indices, seasonally adjusted, for different countries, taken from 1997 to 2007. All the considered countries are members of Organization for Economic Cooperation and Development (OECD), and in particular, this dataset is available from the Statistics Portal of OECD (OECD, 2005). For this dataset, short term forecast, with horizon $b = 1$ (1 month) will be considered. This relatively short horizon makes the weight of the metric shift from the actual forecast (they will be very similar to their last observed point, so the fact that is a forecast is less relevant) to its shape (density). The dataset is show in figure 5.8.

As in previous example, the input data is nonstationary and must undergo some transformation due the the assumptions of the bootstrap method used in the metric. As in the CO_2 dataset, the program TRAMO (Gómez and Maravall, 1996) was used to identify the appropiate transformation for each series. An average linkage hierarchical clustering algorithm was used to generate the dendogram shown in Figure 5.9. The results of the classificacion are compared with the densities depicted in Figure 5.10. Taking a closer look at the case of Spain, Mexico and the United States we can see the effect of a distance based on densities instead of point forecasts. If we take the prediction centers for these three countries, 107.9, 109.3 and 110.1 repectively, the point based distance would group the United States with Mexico before than with Spain, but if we take the whole density of the predicion, it can be argued that in fact Spain and Mexico are closer between them than to the United States, a fact that is correctly reflected in the distance and in the consequent dendogram. At the root of this graph, the two groups that are formed are coherent with the point forecast and the last value of the series, since countries like Greece and Turkey have completely disjoint supports, their L_1 distance is maximum and they are consequently joined in the last step. In this dataset, it can be said that the particular effects of the density based metric are more clear in the initial steps, and tend to behave like a point based metric once the groups get more dissimilar and the respective densities of their members become disjoint.

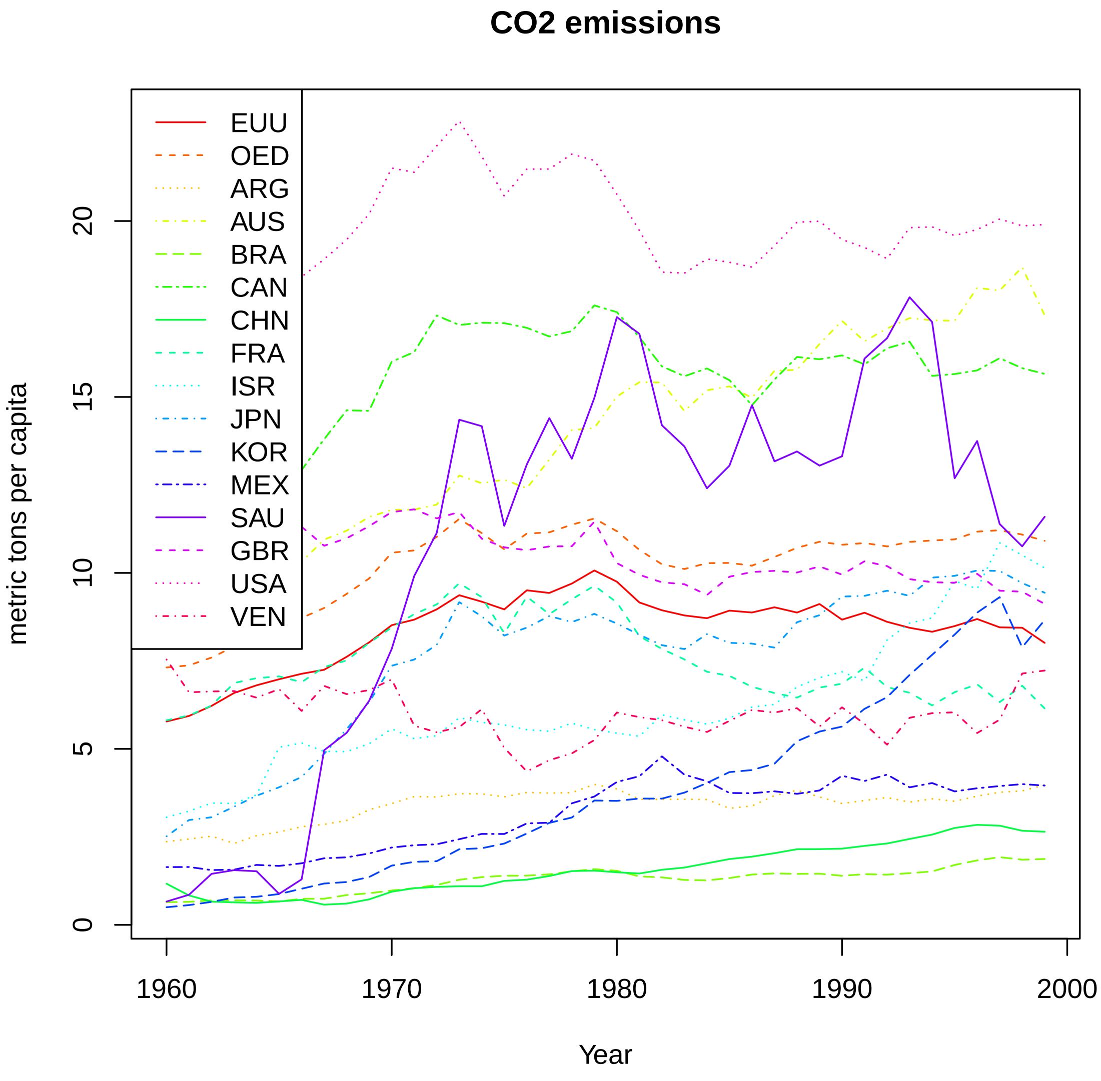


Figure 5.5: Annual CO_2 emissions.

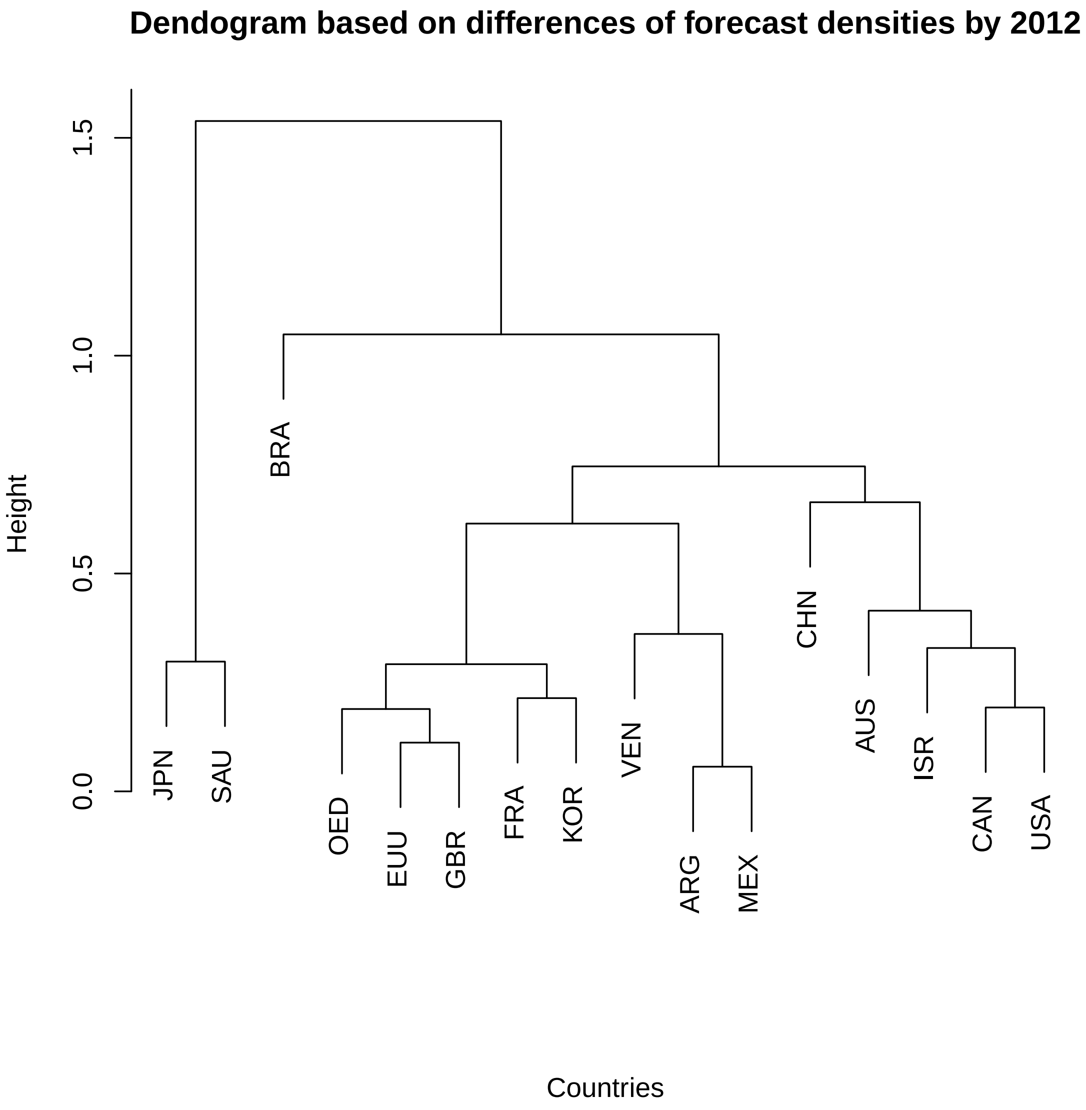


Figure 5.6: Dendogram based on the differences of forecasts densities of CO_2 emissions.

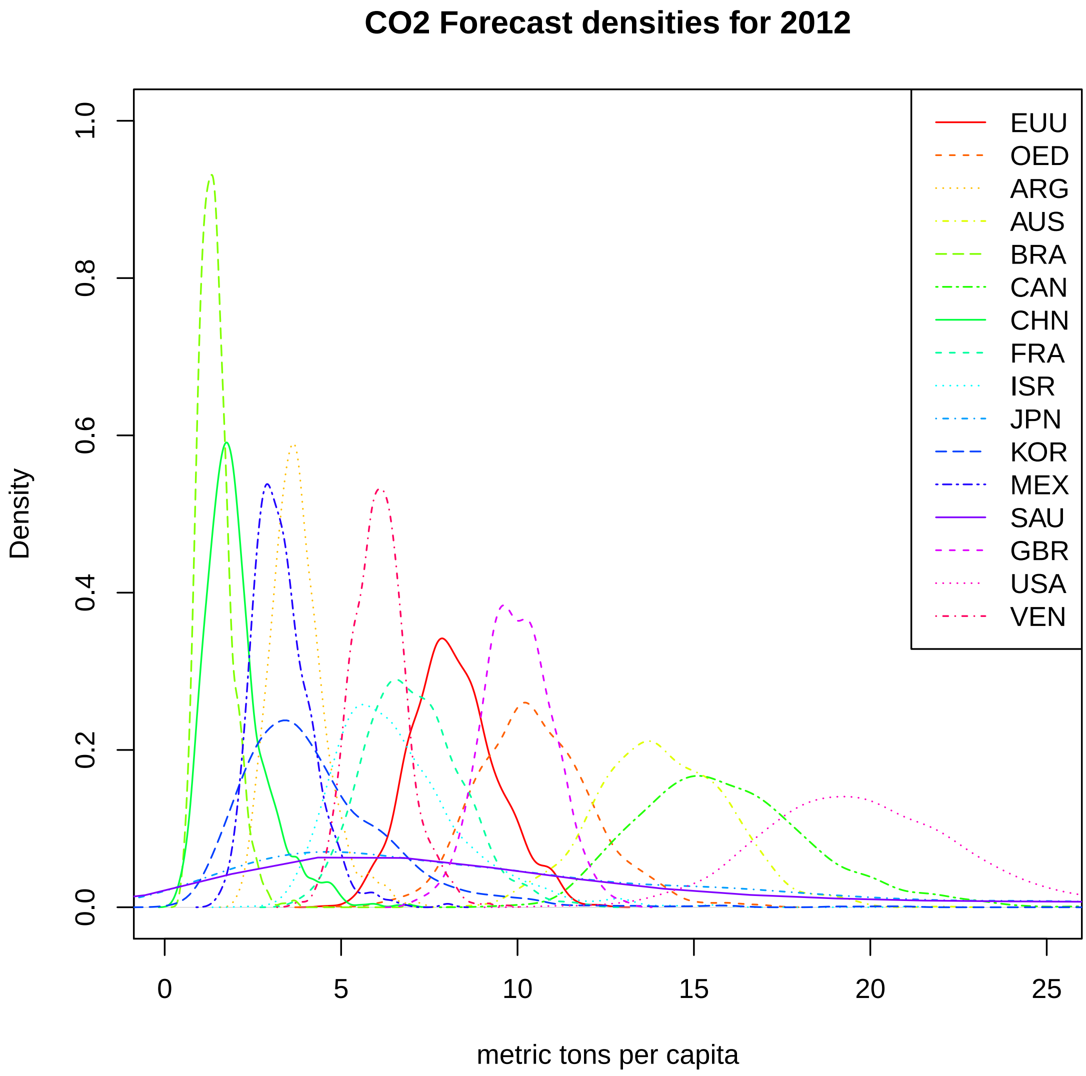


Figure 5.7: Forecasts densities of CO_2 emissions.

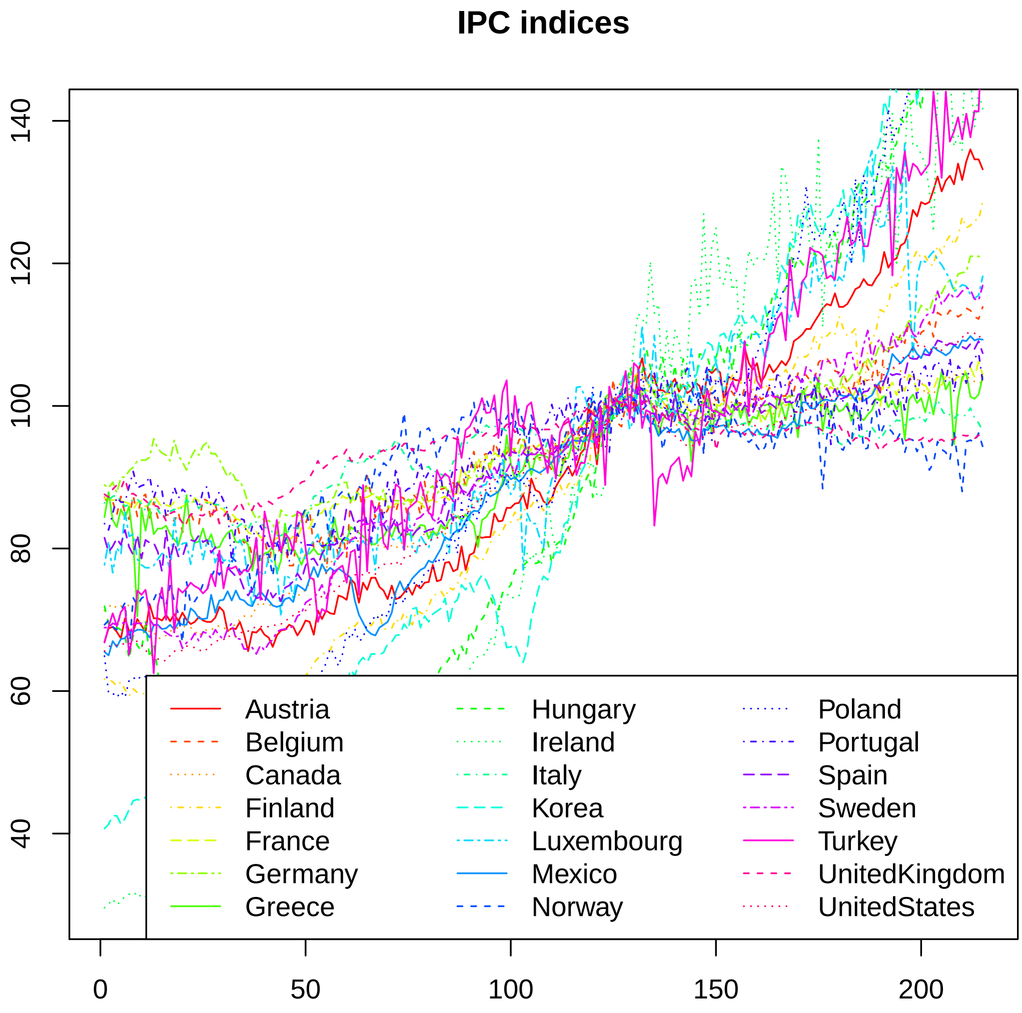


Figure 5.8: IPC indices.

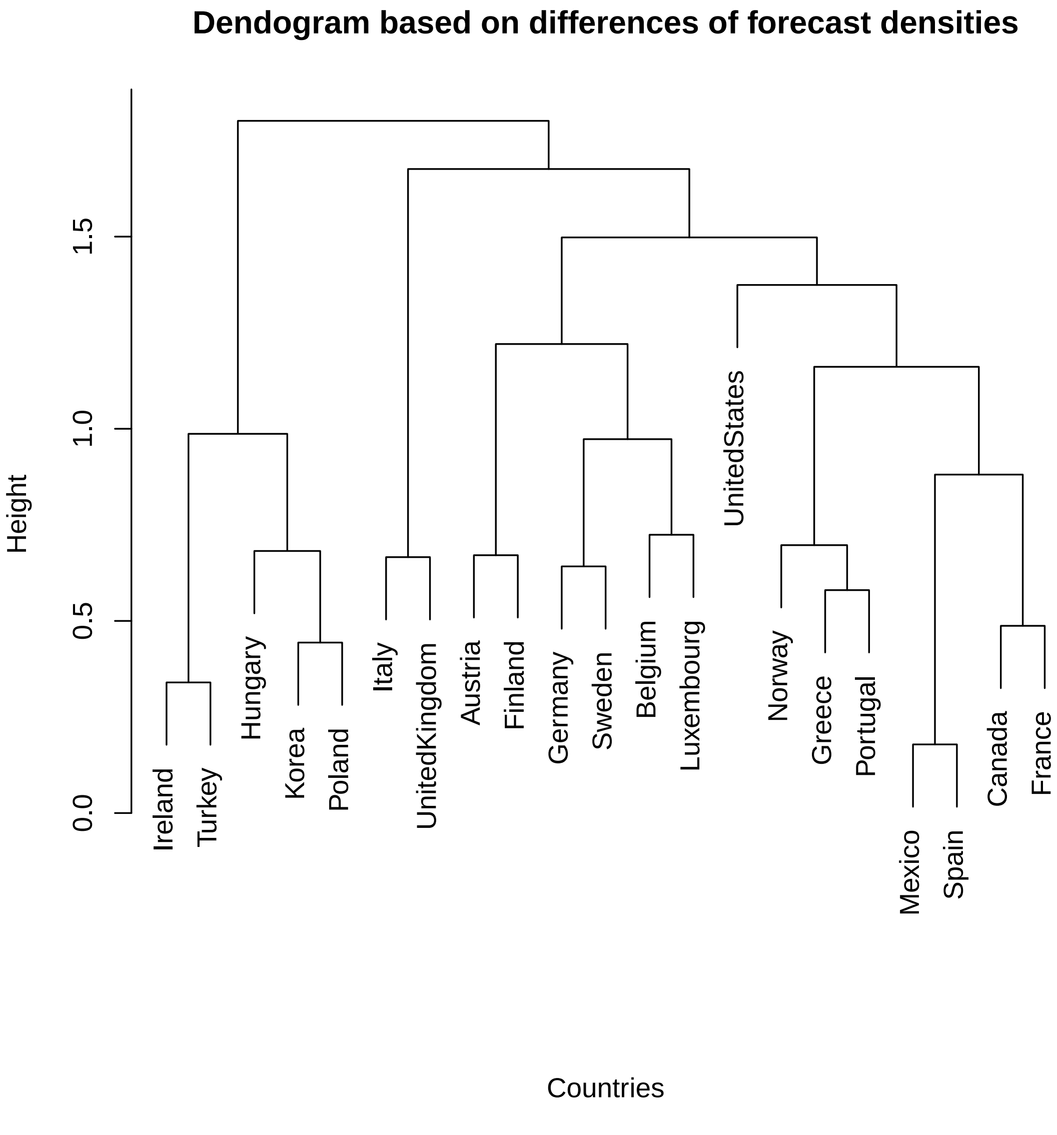


Figure 5.9: Dendogram based on the differences of forecasts densities of IPC indices.

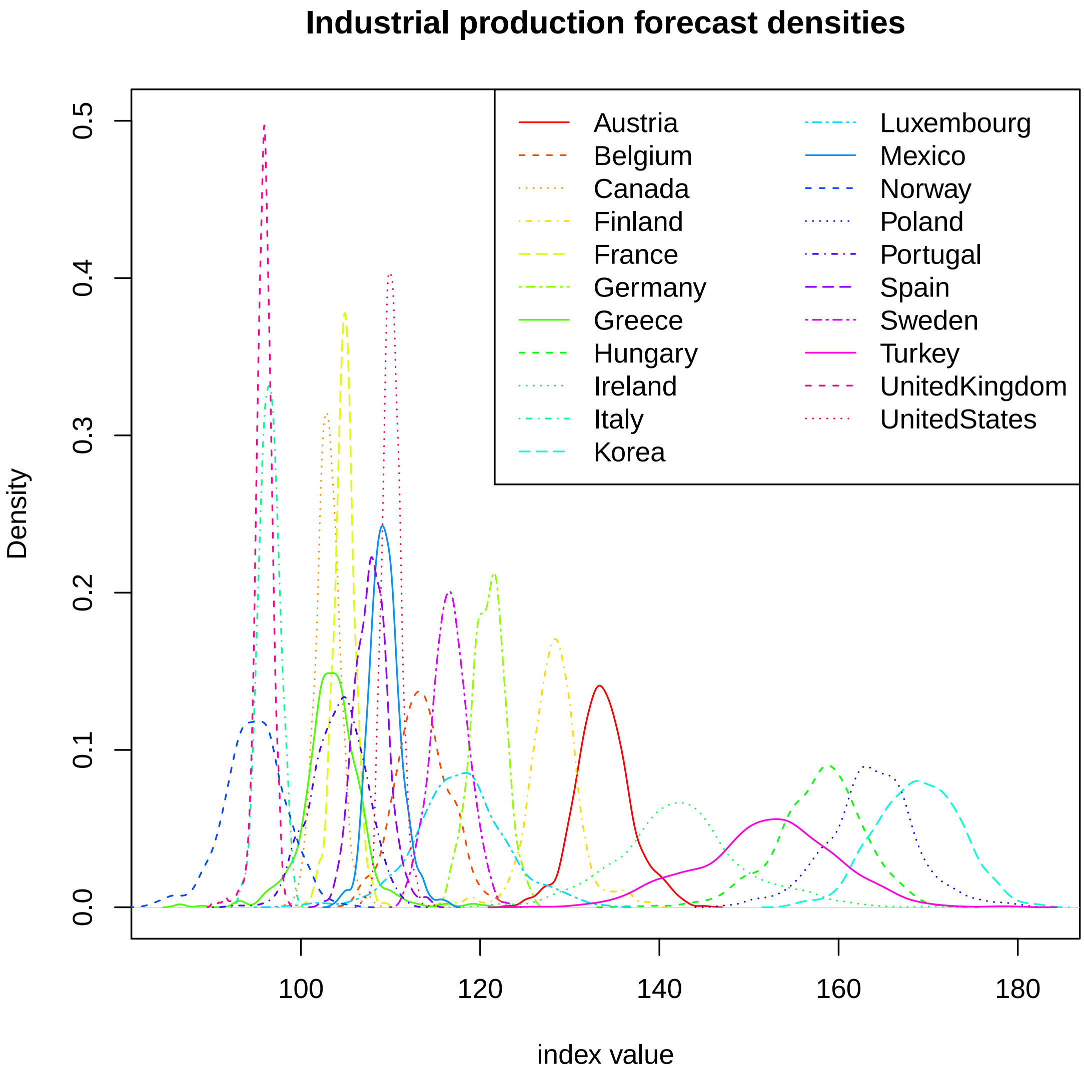


Figure 5.10: Forecasts densities of IPC indices.

Chapter 6

Conclusions

6.1 Design discussion

Since there are many distance metrics that can be applied to time series and with extensibility in mind, a careful consideration of what a time series distance function should be was required. We have opted for the definition of a distance as a function that takes two parameters, the two series to compute the distance, as numeric vectors. This is the simplest definition of a distance between two series (from the software point of view). The main drawback of this approach is that some distances can reuse data or intermediate results when working with the same series, such as the fitted AR models or calculated periodograms and this information cannot be kept if only two series are given as input to the function. On the other hand, if a list or matrix of series is given as a parameter, each distance function must consider these issues, overly complicating its implementation, even when there is no neccessity for it. If computational requirements are costly, an alternate function can be created when needed.

Some distance can take extra parameters besides the two series, such as bandwidths or the order of ARMA models to be fitted, maximum autocorrelation lag to be considered, amount of cepstral coefficients and so on. These parameters have default parameters for a more simple usage.

The auxiliary function `stat_ts_clust` takes a matrix of series and a distance function defined followen the considerations above. This usage is inspired by methods such as `optimize` or `integrate`, that take a function as a parameter. If a distance requires fine grain specification of a parameter, an auxiliary function can be easily defined.

6.2 Usage and Functions Description

See the `statTSclust` package manual included below.

6.3 Conclusions and Future Work

This works presents a series of dissimilarity measures created for the problem of time series clustering, taken from the scientific literature available in this area, that are being implemented as a software package for the popular statistical suite **R**. These metrics are first briefly described, and then their performance is illustrated using a simulation study and some real dataset examples. Since this clustering is an automatic task, special care on lessening the need of human intervention has been taken. There are many proposed dissimilarity measures in the context of time series clustering but the ones presented here compose a reasonable subset of them. The final purpose is the creation of a software package that includes, maintains and becomes a point of reference for the efforts on the implementation of solutions in this area, for the benefit of the scientific community.

Package ‘statTSclust’

January 8, 2013

Type Package

Title Stationary time series clustering

Version 1.0

Date 2013-01-07

Author Pablo Montero Manso, Jose Antonio Vilar

Maintainer `<pmontm@gmail.com>`

Description A package containing functions used in time series clustering: distance metrics and specific clustering methods.

License GPL-2

Depends nlme,locpol, KernSmooth, dtw, longitudinalData

R topics documented:

distance.ACFG 2
distance.ACFU 3
distance.CEPST 4
distance.CHOUAK.DTW 4
distance.CHOUAK.FRECH 5
distance.EUCL 6
distance.GLK 7
distance.INTEGP 8
distance.ISD 9
distance.LNP 10
distance.MAH 11
distance.MAHEXT 12
distance.NP 13
distance.P 14
distance.PACFG 15
distance.PACFU 16
distance.PIC 17

distance.PRED 18
distance.W.LK 18
distance.W.LS 19
maharaj_clust 20
stat_ts_dist 21

Index **22**

distance.ACFG — *Autocorrelation-function based distance with geometrically decaying weights*

Description

Computes the distance based on the autocorrelation function of two given time series. The weight of each autocorrelation coefficient in the final distance is weighted geometrically decreasing.

Usage

```
distance.ACFG(x, y, p = 0.05)
```

Arguments

x	numeric vector containing the first of the two time series.
y	numeric vector containing the second of the two time series.
p	the parameter controlling the decay of each successive autocorrelation coefficient.

Value

The computed AFCG distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

See Also

distance.ACFU, distance.PACFG, distance.PACFU

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.ACFG(x, y, 0.5)
distance.ACFG(x, z, 0.5)
distance.ACFG(y, z, 0.5)
```

distance.ACFU *Autocorrelation-function based distance with uniform weights*

Description

Computes the distance based on the autocorrelation function of two given time series. All the successive autocorrelation coefficients have the same weight in the final distance.

Usage

```
distance.ACFU(x, y)
```

Arguments

x	numeric vector containing the first of the two time series.
y	numeric vector containing the second of the two time series.

Value

The computed ACFU distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

See Also

distance.ACFG, distance.PACFU, distance.PACFG

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.ACFU(x, y)
distance.ACFU(x, z)
distance.ACFU(y, z)
```

`distance.CEPST` *Cepstral coefficients based distance*

Description

Computes the distance based on the cepstral coefficients of two given time series.

Usage

```
distance.CEPST(x, y, k = 50)
```

Arguments

`x`	numeric vector containing the first of the two time series.
`y`	numeric vector containing the second of the two time series.
`k`	the amount of cepstral coefficients to be considered.

Value

The computed distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.CEPST(x, y, 25)
distance.CEPST(x, z, 25)
distance.CEPST(y, z, 25)
```

`distance.CHOUAK.DTW` *Dynamic Time Warping distance weighted by Chouakria-Douzal temporal correlation coefficient*

Description

Computes the distance between time series based on the dynamic time warping distance and the Chouakria-Douzal temporal correlation coefficient.

Usage

```
distance.CHOUAK.DTW(x, y, k = 1)
```

Arguments

x	numeric vector containing the first of the two time series.
y	numeric vector containing the second of the two time series.
k	parameter controlling the contribution between the dynamic time warping distance and the Chouakria-Douzal temporal correlation coefficient. k must be greater or equal to 0.

Value

The computed distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

See Also

distance.CHOUAK.FRECH

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.CHOUAK.DTW(x, y, 1)
distance.CHOUAK.DTW(x, z, 1)
distance.CHOUAK.DTW(y, z, 1)
```

distance.CHOUAK.FRECH *Frechet distance weighted by Chouakria-Douzal temporal correlation coefficient*

Description

Computes the distance between time series based on the Frechet distance and the Chouakria-Douzal temporal correlation coefficient.

Usage

```
distance.CHOUAK.FRECH(x, y, k = 1)
```

Arguments

x	numeric vector containing the first of the two time series.
y	numeric vector containing the second of the two time series.
k	parameter controlling the contribution between the Frechet and the Chouakria-Douzal temporal correlation coefficient. k must be greater or equal to 0.

Value

The computed distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

See Also

distance.CHOUAK.DTW

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.CHOUAK.FRECH(x, y, 1)
distance.CHOUAK.FRECH(x, z, 1)
distance.CHOUAK.FRECH(y, z, 1)
```

distance.EUCL *Euclidean distance*

Description

Computes the Euclidean distance between two time series.

Usage

```
distance.EUCL(x, y)
```

Arguments

x	numeric vector containing the first of the two time series.
y	numeric vector containing the second of the two time series.

Details

This distance is included as an auxiliary function, having the same usage as the others distance metrics include in this package.

Value

The computed distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.EUCL(x, y)
distance.EUCL(x, z)
distance.EUCL(y, z)
```

`distance.GLK` *Generalized Likelihood Ratio Test based distance*

Description

Computes the distance based on the generalized likelihood ratio test between the spectra of two time series

Usage

```
distance.GLK(x, y)
```

Arguments

x	numeric vector containing the first of the two time series.
y	numeric vector containing the second of the two time series.

Details

High computational requirements.

Value

The computed distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

See Also

distance.ISD

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.GLK(x, y)
distance.GLK(x, z)
distance.GLK(y, z)
```

distance.INTEGP *Integrated Periodogram based distance*

Description

Computed the distance between two time series based on their integrated periodograms.

Usage

```
distance.INTEGP(x, y)
```

Arguments

x	numeric vector containing the first of the two time series.
y	numeric vector containing the second of the two time series.

Value

The computed distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.INTEGP(x, y)
distance.INTEGP(x, z)
distance.INTEGP(y, z)
```

`distance.ISD` *Integrated Squared Differences between log-spectra distance*

Description

Computes the distance between two time series based on the integrated squared differences between the non-parametric estimators of their log-spectra.

Usage

```
distance.ISD(x, y)
```

Arguments

`x`	numeric vector containing the first of the two time series.
`y`	numeric vector containing the second of the two time series.

Value

The computed distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

See Also

`distance.GLK`

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.ISD(x, y)
distance.ISD(x, z)
distance.ISD(y, z)
```

distance.LNP *Logarithmic Normalized Periodogram distance*

Description

Computes the distance between two time series based on the logarithm of their normalized periodograms.

Usage

```
distance.LNP(x, y)
```

Arguments

x	numeric vector containing the first of the two time series.
y	numeric vector containing the second of the two time series.

Value

The computed distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

See Also

distance.P, distance.NP

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.LNP(x, y)
distance.LNP(x, z)
distance.LNP(y, z)
```

distance.MAH *Maharaj distance*

Description

Computes the distance between two time series based on the test on the homogeneity of generating ARMA models.

Usage

```
distance.MAH(x, y)
```

Arguments

x	numeric vector containing the first of the two time series.
y	numeric vector containing the second of the two time series.

Details

The ARMA models are fitted automatically and the degree selected by the AIC criterion.

Value

The computed distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

See Also

`distance.MAHEXT`, `distance.PIC`

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.MAH(x, y)
distance.MAH(x, z)
distance.MAH(y, z)
```

distance.MAHEXT *Extended Maharaj distance*

Description

Computes the distance between two time series based on their AR coefficients when the series are fitted by ARMA models and extended by the inclusion of the correlation coefficient between the series.

Usage

```
distance.MAHEXT(x, y, k)
```

Arguments

x	numeric vector containing the first of the two time series.
y	numeric vector containing the second of the two time series.
k	degree of the AR model to be fitted.

Value

The computed distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

See Also

distance.MAH, distance.PIC

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.MAHEXT(x, y, 2)
distance.MAHEXT(x, z, 2)
distance.MAHEXT(y, z, 2)
```

distance.NP *Normalized Periodogram distance*

Description

Computes the distance between two time series based on their normalized periodograms.

Usage

```
distance.NP(x, y)
```

Arguments

x	numeric vector containing the first of the two time series.
y	numeric vector containing the second of the two time series.

Value

The computed distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

See Also

distance.P, distance.LNP

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.NP(x, y)
distance.NP(x, z)
distance.NP(y, z)
```

distance.P *Periodogram based distance*

Description

Computes the distance between two time series based on their periodograms.

Usage

```
distance.P(x, y)
```

Arguments

x	numeric vector containing the first of the two time series.
y	numeric vector containing the second of the two time series.

Value

The computed distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

See Also

distance.NP, distance.LNP

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.P(x, y)
distance.P(x, z)
distance.P(y, z)
```

distance.PACFG *Partial Autocorrelation function based with geometric decaying weights distance*

Description

Computes the distance based on the partial autocorrelation function of two given time series. The weight of each partial autocorrelation coefficient in the final distance is weighted geometrically decreasing.

Usage

```
distance.PACFG(x, y, p = 0.05)
```

Arguments

x	numeric vector containing the first of the two time series.
y	numeric vector containing the second of the two time series.
p	the parameter controlling the decay of each successive autocorrelation coefficient.

Value

The computed distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

See Also

distance.ACFU, distance.ACFG, distance.PACFU

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.PACFG(x, y, 0.5)
distance.PACFG(x, z, 0.5)
distance.PACFG(y, z, 0.5)
```

distance.PACFU *Partial Autocorrelation function based distance with uniform weights*

Description

Computes the distance based on the partial autocorrelation function of two given time series. All the successive partial autocorrelation coefficients have the same weight in the final distance.

Usage

```
distance.PACFU(x, y)
```

Arguments

x	numeric vector containing the first of the two time series.
y	numeric vector containing the second of the two time series.

Value

The computed distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

See Also

distance.PACFG, distance.ACFU, distance.ACFG

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.PACFU(x, y)
distance.PACFU(x, z)
distance.PACFU(y, z)
```

`distance.PIC` *Piccolo distance*

Description

Computes the distance between two time series based on their AR coefficients when the series are fitted by ARIMA models.

Usage

```
distance.PIC(x, y)
```

Arguments

`x`	numeric vector containing the first of the two time series.
`y`	numeric vector containing the second of the two time series.

Details

The ARMA models are fitted automatically and the degree selected by the AIC criterion.

Value

The computed distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

See Also

`distance.MAH`, `distance.MAHEXT`

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.PIC(x, y)
distance.PIC(x, z)
distance.PIC(y, z)
```

`distance.PRED` *Nonparametric forecast based distance*

Description

Computes the distance between two time series based on their bootstrap density of their forecasts at a given horizon.

Usage

```
distance.PRED(x, y, k = 5)
```

Arguments

`x`	numeric vector containing the first of the two time series.
`y`	numeric vector containing the second of the two time series.
`k`	the forecast horizon in time steps after the end of the given series.

Value

The computed distance.

Author(s)

Jose Antonio Vilar, Pablo Montero Manso.

Examples

```
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.PRED(x, y, 5)
distance.PRED(x, z, 5)
distance.PRED(y, z, 5)
```

`distance.W.LK` *Spectral dissimilarity distance via maximum likelihood estimation of the log-spectra*

Description

Computed the distance between two time series based on the spectral dissimilarity between their spectral densities estimated by the exponentiation of the estimation by maximum likelihood of their log-spectra.

Usage

```
distance.W.LK(x, y)
```

Arguments

x	numeric vector containing the first of the two time series.
y	numeric vector containing the second of the two time series.

Value

The computed distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

See Also

distance.W.LS

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.W.LK(x, y)
distance.W.LK(x, z)
distance.W.LK(y, z)
```

---

distance.W.LS *Spectral dissimilarity distance via least squares estimation of the log-spectra*

Description

Computed the distance between two time series based on the spectral dissimilarity between their spectral densities estimated by the exponentiation of the estimation by lest squares of their log-spectra.

Usage

```
distance.W.LS(x, y)
```

Arguments

x	numeric vector containing the first of the two time series.
y	numeric vector containing the second of the two time series.

Value

The computed distance.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

See Also

distance.W.LK

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
distance.W.LS(x, y)
distance.W.LS(x, z)
distance.W.LS(y, z)
```

---

maharaj_clust *Maharaj p.value-based Clustering Algorithm*

Description

Computes the distance based on the autocorrelation function of two given time series. The weight of each autocorrelation coefficient in the final distance is weighted geometrically decreasing.

Usage

```
maharaj_clust(distances, significance)
```

Arguments

distances	a dist object containing the distances between the series to cluster.
significance	the significance level.

Value

A list with the indices of the series. Each element of the list is a numeric vector with the indices of the series that are grouped into the same cluster.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

See Also

stat_ts_dist

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
## Compute the distance and check for coherent results
dd = stat_ts_dist( rbind(x,y,z), distance.MAH)
maharaj_clust( dd, 0.05 )
```

stat_ts_dist *Distance between time series*

Description

Computes the `dist` object for the given series using the given distance metric.

Usage

```
stat_ts_dist(series, dist_fun)
```

Arguments

`series`	matrix containing the series in row order
`dist_fun`	A distance function that takes two series (x,y) as parameters.

Value

A list with the indices of the series. Each element of the list is a numeric vector with the indices of the series that are grouped into the same cluster.

Author(s)

Pablo Montero Manso, Jose Antonio Vilar.

Examples

```
## Create three sample time series
x <- cumsum(rnorm(100))
y <- cumsum(rnorm(100))
z <- sin(seq(0, pi, length.out=100))
dd = stat_ts_dist( rbind(x,y,z), distance.EUCL)
```

Index

∗Topic **\textasciitildekwd1**
- distance.ACFG, 2
- distance.ACFU, 3
- distance.CEPST, 4
- distance.CHOUAK.DTW, 4
- distance.CHOUAK.FRECH, 5
- distance.EUCL, 6
- distance.GLK, 7
- distance.INTEGP, 8
- distance.ISD, 9
- distance.LNP, 10
- distance.MAH, 11
- distance.MAHEXT, 12
- distance.NP, 13
- distance.P, 14
- distance.PACFG, 15
- distance.PACFU, 16
- distance.PIC, 17
- distance.PRED, 18
- distance.W.LK, 18
- distance.W.LS, 19
- maharaj_clust, 20
- stat_ts_dist, 21

∗Topic **\textasciitildekwd2**
- distance.ACFG, 2
- distance.ACFU, 3
- distance.CEPST, 4
- distance.CHOUAK.DTW, 4
- distance.CHOUAK.FRECH, 5
- distance.EUCL, 6
- distance.GLK, 7
- distance.INTEGP, 8
- distance.ISD, 9
- distance.LNP, 10
- distance.MAH, 11
- distance.MAHEXT, 12
- distance.NP, 13
- distance.P, 14
- distance.PACFG, 15
- distance.PACFU, 16
- distance.PIC, 17
- distance.PRED, 18
- distance.W.LK, 18
- distance.W.LS, 19
- maharaj_clust, 20
- stat_ts_dist, 21

distance.ACFG, 2, *3*, *15*, *16*
distance.ACFU, *2*, 3, *15*, *16*
distance.CEPST, 4
distance.CHOUAK.DTW, 4, *6*
distance.CHOUAK.FRECH, 5, *5*
distance.EUCL, 6
distance.GLK, 7, *9*
distance.INTEGP, 8
distance.ISD, *8*, 9
distance.LNP, 10, *13*, *14*
distance.MAH, 11, *12*, *17*
distance.MAHEXT, *11*, 12, *17*
distance.NP, *10*, 13, *14*
distance.P, *10*, *13*, 14
distance.PACFG, *2*, *3*, 15, *16*
distance.PACFU, *2*, *3*, *15*, 16
distance.PIC, *11*, *12*, 17
distance.PRED, 18
distance.W.LK, 18, *20*
distance.W.LS, 19, *19*

maharaj_clust, 20

stat_ts_dist, 21, *21*

Bibliography

Andrés M. Alonso, J.R. Berrendero, A. Hernandez, and A. Justel. Time series clustering based on forecast densities. *Comput. Statist. Data Anal.*, 51:762–776, 2006. ISSN 0167-9473.

P. Berkhin. A survey of clustering data mining techniques. *Grouping multidimensional data*, pages 25–71, 2006.

Jorge Caiado, Nuno Crato, and Daniel Peña. A periodogram-based metric for time series classification. *Comput. Statist. Data Anal.*, 50(10):2668–2684, 2006. ISSN 0167-9473.

D. Casado de Lucas. Classification techniques for time series and functional data. 2010.

A. Chouakria-Douzal and P. N. Nagabhushan. Adaptive dissimilarity index for measuring time series proximity. *Adv. Data Anal. Classif.*, 1(1):5–21, 2007. ISSN 1862-5347.

Maercella Corduas and Domenico Piccolo. Time series clustering and classification by the autoregressive metric. *Comput. Statist. Data Anal.*, 52(4):1860–1872, 2008. ISSN 0167-9473.

T. Eiter and H. Mannila. Computing discrete fréchet distance. *See Also*, 1994.

B.S. Everitt, S. Landau, M. Leese, et al. Cluster analysis, 2001.

Jianqing Fan and Eva Kreutzberger. Automatic local smoothing for spectral density estimation. *Scand. J. Statist.*, 25(2):359–369, 1998. ISSN 0303-6898.

Jianqing Fan and Wenyang Zhang. Generalised likelihood ratio tests for spectral density. *Biometrika*, 91(1):195–209, 2004. ISSN 0006-3444.

M.M. Fréchet. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 22(1):1–72, 1906.

J.K. Galbraith and L. Jiaqing. Cluster and discriminant analysis on time series as a research tool. In *UTIP Working Paper Number 6.* Austin: Lyndon B. Johnson School fo Public Affairs. The University of Texas at Austin, 1999.

Pedro Galeano and Daniel Peña. Multivariate analysis in vector time series. *Resenhas*, 4(4):383–403, 2000. ISSN 0104-3854.

Martin Gavrilov, Dragomir Anguelov, Piotr Indyk, and Rajeev Motwani. 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, pages 487–496, New York, USA, 2000. ACM.

A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000 (June 13). Circulation Electronic Pages: http://circ.ahajournals.org/cgi/content/full/101/23/e215 PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.

V. Gómez and A. Maravall. 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, 1996. URL `www.bde.es`.

P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Mathematical programming*, 79(1):191–215, 1997.

B. Hirsch and D. DuBois. Self-esteem in early adolescence: the identification and prediction of contrasting longitudinal trajectories. 20(1):53–72, 1991.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *J. Classification*, 2: 193–218, 1985.

A.K. Jain and R.C. Dubes. *Algorithms for clustering data.* Prentice-Hall, Inc., 1988.

A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

Yoshihide Kakizawa, Robert H. Shumway, and Masanobu Taniguchi. Discrimination and clustering for multivariate time series. *J. Amer. Statist. Assoc.*, 93(441):328–340, 1998. ISSN 0162-1459.

Konstantinos Kalpakis, D. Gada, and Vasundhara Puttagunta. Distance measures for effective clustering of arima time-series. In Nick Cercone, Tsau Young Lin, and XindongEditors Wu, editors, *Proceedings 2001 IEEE International Conference on Data Mining*, pages 273–280. IEEE Comput. Soc, 2001.

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

Z.J. Kovačić. 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*, pages 204–207. Springer, 1998.

C. Li, G. Biswas, M. Dale, and P. Dale. Building models of ecological dynamics using hmm based temporal data clustering. In *Proc. of the Fourth International Conference on Intelligent Data Analysis*, 2001.

T. Warren Liao. Clustering of time series data : a survey. *Pattern Recognition*, 38 (11):1857–1874, 2005.

Martin Maechler, Peter Rousseeuw, Anja Struyf, Mia Hubert, and Kurt Hornik. *cluster: Cluster Analysis Basics and Extensions*, 2011. R package version 1.14.1 — For new features, see the 'Changelog' file (in the package source).

Elizabeth Ann Maharaj. A significance test for classifying ARMA models. *J. Statist. Comput. Simulation*, 54(4):305–331, 1996. ISSN 0094-9655.

Elizabeth Ann Maharaj. Clusters of time series. *J. Classification*, 17(2):297–314, 2000. ISSN 0176-4268.

Elizabeth Ann Maharaj. Comparison of non-stationary time series in the frequency domain. *Comput. Statist. Data Anal.*, 40(1):131–141, 2002. ISSN 0167-9473.

T. Oates, L. Firoiu, and P.R. Cohen. Clustering time series with hidden markov models and dynamic time warping. In *Proceedings of the IJCAI-99 workshop on neural, symbolic and reinforcement learning methods for sequence learning*, pages 17–21. Citeseer, 1999.

OECD. Oecd data. `http://stats.oecd.org/`, 2005.

Sonia Pértega and José Antonio Vilar. Comparing several parametric and nonparametric approaches to time series clustering: A simulation study. *J. Classification*, 27(3):333–362, November 2010. ISSN 0176-4268.

Domenico Piccolo. A distance measure for classifying arima models. *J. Time Series Anal.*, 11(2):153–164, 1990. ISSN 1467-9892.

R Development Core Team. *A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria, 2004. URL `www.R-project.org`.

M. Ramoni, P. Sebastiani, and P. Cohen. Bayesian clustering by dynamics. *Machine learning*, 47(1):91–121, 2002.

William M. Rand. Objective Criteria for the Evaluation of Clustering Methods. *J. Amer. Statist. Assoc.*, 66(336):846–850, 1971.

Peter Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, 1987.

D. Ruppert, S.J. Sheather, and M.P. Wand. An effective bandwidth selector for local least squares regression. *Journal of the American Statistical Association*, 90(432): 1257–1270, 1995.

Allou Samé, Faicel Chamroukhi, Gérard Govaert, and Patrice Aknin. Model-based clustering and segmentation of time series with changes in regime. *Adv. Data Anal. Classif.*, 5(4):301–321, 2011. ISSN 1862-5347.

D. Sankoff and J.B. Kruskal. Time warps, string edits, and macromolecules: the theory and practice of sequence comparison. *Reading: Addison-Wesley Publication, 1983, edited by Sankoff, David; Kruskal, Joseph B.*, 1, 1983.

P. Smyth et al. Clustering sequences with hidden markov models. *Advances in neural information processing systems*, pages 648–654, 1997.

Zbigniew R. Struzik and Arno Siebes. 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*, pages 12–22. Springer, 1999.

Howell Tong and P Dabas. Cluster of time series models: an example. 17(2):187–198, 1990. doi: 10.1080/757582830.

Howell Tong and Iris Yeung. On tests for self-exciting threshold autoregressive-type non-linearity in partially observed time series. *Appl. Statist.*, 40(1):43–62, 2000. ISSN 00359254.

José Antonio Vilar and Sonia Pértega. Discriminant and cluster analysis for gaussian stationary processes: local linear fitting approach. *J. Nonparametr. Stat.*, 16(3-4): 443–462, 2004. ISSN 1048-5252.

José Antonio Vilar, Andrés M. Alonso, and Juan Manuel Vilar. Non-linear time series clustering based on non-parametric forecast densities. *Comput. Statist. Data Anal.*, 54(11):2850–2865, November 2010. ISSN 0167-9473.

Juan Manuel Vilar, José Antonio Vilar, and Sonia Pértega. Classifying time series data: A nonparametric approach. *J. Classification*, 26(1):3–28, April 2009. ISSN 0176-4268.

R. Xu, D. Wunsch, et al. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.