Clustering of Time Series using Visibility Graphs and Quantile Graphs

Vanessa Silva
Faculty of Sciences, University of Porto
Porto, Portugal
vanessa.silva@dcc.fc.up.pt

***Abstract*—Mining interesting features from time series is a crucial task for time series clustering. Typically, such features are obtained from time series specific characteristics such as trend, period, seasonality and other global measures. A recent approach consists on mapping the time series into a graph and then characterize the time series from a graph point of view, that is, using topological metrics. This approach depends on the mapping of the series onto the graph, which is not always an obvious task. In this work two concepts of existing mappings are explored and it is shown that the joint use of these mappings can be an advantage for the grouping of time series. In order to evaluate the proposed approach, the time series grouping based on the metrics extracted from the visibility networks and the quantile networks of time series is applied to a set of specific time series models. The results are promising and show the networks' potential for time series grouping.**

***Index Terms*—Clustering, Time Series, Complex Networks, Topological Features**

I. Introduction

A time series is a collection of observations indexed in time. The main purpose of time series analysis is to develop mathematical models that provide plausible descriptions of the characteristics of the data with a view to forecasting, simulation and control [1].

The classification of time series is an intrinsic activity to facilitate the handling and the organization of the enormous amount of information that we can capture. However, this is still a very explored field given the great diversity of data and the difficulty in finding an ideal model for the accomplishment of such series classification [2].

The analysis of complex networks has been receiving increasing interest from the research community [3] and this led to the emergence of the new field of Network Science [4]. This field has shown to be very promising with respect to data clustering tasks [5], through the use of topological graph measurements that are currently available [6].

Several network-based time series analysis approaches have been recently proposed, based on mapping time series to the network domain. The mappings proposed in the literature are based on concepts such as correlation [7], phase space reconstruction [8], recurrence analysis [9], visibility [10] or transition probabilities [11]. Some mappings result in networks that have as many nodes as the number of observations in the time series, such as visibility mappings, but others, such as a quantile based mapping [11], allow to reduce the dimensionality of the series while preserving the characteristics of the time dynamics. Network-based time series analysis techniques have been showing promising results and have been successful in the description, classification and clustering of time series of real datasets. Examples of this include automatic classification of sleep stages [12], characterizing the dynamics of human heartbeat [13], distinguishing healthy from non-healthy electroencephalographic (EEG) series [14] and analyzing seismic signals [15].

The general problem of time series clustering concerns the separation of a set of time series into clusters, with the property that the series of the same group have a similar structure and characteristics, and different from the series of other groups. A fundamental problem in the clustering and classification analysis is the choice of a relevant metric. Existing time series clustering approaches are mainly based on methods of distance, such as the Euclidean distance in space points in order to separate the group of time series of clusters, and on approaches based on resources extracted in time domain, frequency domain and wavelet decomposition of the time series, which are later grouped using grouping methods [16]. These approaches have limitations, distance-based methods do not produce the best results and proving to be insufficient. And the methods based on the characteristics of the series depends on the calculation of these characteristics which is not a trivial task since there are several ways to reach your result (or its approximation), as well as the need to pay attention to some parameters, given the diversity of existing time series.

In this work we propose a new approach to group time series in different classes. This approach consists of constructing three network mapping (natural visibility graphs, horizontal visibility graphs and quantile graphs), for each of the time series and calculating the global topological metrics of these networks, that is, average grade, average path length, number of communities, clustering coefficient, and modularity. This set of metrics forms a vector of topological characteristics of the networks that served to feed a clustering algorithm, k-means.

We show the potential of this new approach in a large set of simulated time series models, linear and nonlinear models, and confirm that the use of the different types of mapping of these models in networks results in a set of features that can capture information encoded in each one of the models and thus distinguish them from unsupervised manner.

This paper is organized as follows. We start, in the section II, with a presentation of the background on time series

and complex networks. In the section III, we presented a background of time series mappings to complex networks, in the section IV we describe of the approach proposed for this work and presented of the obtained results and their analysis. Finally, in the last section (V), we presented the main conclusions and we mention some of future work.

II. Basic Concepts

A. Time Series

A time series $Y_T = (y_1, \ldots, y_T)$ is a set of observations collected at different (usually equidistant) points in time. The main characteristic of a time series is the serial dependence between the observations which restricts the applicability of many conventional statistical models traditionally dependent of the assumption of independent and identically distributed (i.i.d) observations. The main purpose of the analysis of a time series is to develop mathematical models that provide plausible descriptions of the characteristics of the data with a view to forecasting, simulation and control [1].

In our work, we are essentially interested in extracting the most relevant characteristics of the time series through the science of networks so that we can distinguish different models of time series. The traditional approach to analyses the data under these circumstances is to decompose the time series in *trend*, *cycle*, *seasonality* and *random components*. Trend component indicates the long term behavior of the series. The cycle is characterized by smooth and repeated oscillations of rise and fall in the time series around the trend. Seasonal component corresponds to the oscillations of ascent and descent that occur with a fixed period, for example, within a year that they are usually related to the seasons of the year. And the random component represents all the other effects resulting from a multiplicity of factors and of unpredictable nature.

We say that a time series is stationary when its statistical characteristics (mean, variance) are constant over time, that is, its data oscillate around a constant mean with the variance of the fluctuations remaining essentially the same. The stationarity implies that correlation between observations depends only on the time lag between the observations. Most of the methods and models used in time series area imply that the series are stationary.

1) Time Series Models: There are many time series models available in the literature. The models are classified into linear and nonlinear. In this work we will simulate a large set of some time series models most used and useful in series analysis. Below we present with more detail each one of these models to use.

The simplest time series process is the purely random process or *white noise*, (ϵ_t). A white noise is a sequence of i.i.d. random variables with zero mean and constant variance, σ_ϵ^2. A particular case is the Gaussian white noise, where ϵ_t are independent normal random variables [17].

a) Linear models: Linear time series models are essentially models for which the conditional mean is a linear function of past values of the time series [18].

AR(p)**:** y_t is an autoregressive process of order p if it satisfies the following equation [17]:

$$y_t = \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t$$

where p is the number of autoregressive terms and ϵ_t is a white noise process. This model explicitly specifies a linear relationship between the current and past values as suggested by its name. In this work we will study three particular AR models: two AR(1) models with $\phi_1 \in \{-0.5, 0.5\}$ and one AR(2) model with $\phi_1 = 1.5$ and $\phi_2 = -0.75$.

ARIMA(p, d, q)**:** an autoregressive moving average, ARMA, process combines AR processes and Moving Average, MA, processes which consist of a linear combination of i.i.d random variables (white noise) [17]. Thus, y_t is an ARMA process of order (p, q) if it satisfies the equation:

$$\begin{aligned} y_t &= \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t \\ \left(1 - \sum_{i=1}^{p} \phi_i B^i\right) y_t &= (1 + \sum_{i=1}^{q} \theta_i B^i)\epsilon_t \\ \Phi(B) y_t &= \Theta(B)\epsilon_t \end{aligned} \quad (1)$$

where the white noise ϵ_t is usually a Gaussian process, ϕ_i, $i = 1, \ldots, p$ are constants such that $\Phi(z) = 1 - \sum_{i=1}^{p} \phi_i z^i \neq 0$ for $|z| \leq 1$ and θ_i, $i = 1, \ldots, q$ are constants such that $\Theta(z) = 1 + \sum_{i=1}^{q} \theta_i z^i \neq 0$ for $|z| \leq 1$. B represents the backshift operator, $By_t = y_{t-1}$.

Now, assume that you have a nonstationary time series, x_t but whose dth-difference $y_t = \nabla^d x_t$ is a stationary ARMA(p, q) process. This means that the time series y whose value at time t is the difference between x_{t+d} and x_t. Then x_t is said an AutoRegressive Integrated Moving Average, ARIMA(p, d, q), process and satisfies the following equation:

$$\Phi(B) y_t (1 - B)^d x_t = \Theta(B)\epsilon_t \quad (2)$$

In this work we will study an ARIMA(1,1,0) model.

ARFIMA(p, d, q)**:** a generalization of the ARIMA(p, d, q) process by allowing the parameter d to assume real values. An time series is said to be a autoregressive fractionally integrated moving average, ARFIMA, model if it satisfies the equation:

$$(1 - \sum_{i=1}^{p} \phi_i B^i)(1 - B)^d x_t = (1 + \sum_{i=1}^{q} \theta_i B^i) + \epsilon_t \quad (3)$$

where $(1 - B)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-B)^k$, parameter d is said the long memory parameter since it controls the rate of decay of the autocorrelation function. When $d \neq 0$ the rate of decay is hyperbolic meaning that persistence in the autocorrelations: there is significant dependence between observations separated by long time intervals [17]. In this work we will study two ARFIMA$(1, 0.4, 0)$ models with $\phi_1 \in \{-0.5, 0.5\}$.

b) Nonlinear models: The initial development of nonlinear time series analysis focused on several nonlinear parametric forms. We can distinguish specifications for the conditional mean and specifications for the conditional variance [19].

SETAR(1)**:** the self-exciting threshold autoregressive (SETAR) models of order 1 specify the nonlinearity in the conditional mean. These are very useful for processes in which regime changes occur, where the idea is to approximate a nonlinear function in a linear function dependent on the regime that changes according to the process values [20]. This model can be presented as follows:

$$y_t = \begin{cases} \alpha y_{t-1} + \epsilon_t, & if \quad y_{t-1} \leq r \\ \beta y_{t-1} + \gamma \epsilon_t, & if \quad y_{t-1} > r \end{cases} \tag{4}$$

where r represents a real threshold and d is the delay parameter.

INAR(1)**:** the INAR models have been proposed to model correlated integer-valued time series [21]. These models are based on thinning (random) operations defined on the integers. The most common such operation is the binomial thinning defined as follows. Let X be an integer valued random variable and $0 < \alpha < 1$. Then $\alpha * X = \sum_{i=1}^{X} \chi_i$ where $\chi_i \sim Be(\alpha)$, meaning that $\alpha * X | X \sim Bi(X, \alpha)$. The time series y_t is said an INAR(1) if it satisfies:

$$y_t = \alpha * y_{t-1} + \epsilon_t \tag{5}$$

where $\alpha \in [0, 1]$, ϵ_t are integer valued time series and $*$ defines the binomial thinning operation.

GARCH(p, q)**:** an autoregressive conditional heteroscedastic model was created to model the volatility (or conditional variance) that is not constant in time, in a homogeneous time model. The basic idea of ARCH models is that the asset return is serially uncorrelated, but dependent, and that the dependence can be described by a simple quadratic function of its lagged values [22]. GARCH is a generalization of the ARCH(q) process that propose that conditional volatility be a function not only of the squares of past errors, which is the case of ARCH models, but also of their own past values [18]. To model a time series σ_t^2 using a generalized ARCH process, we define:

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 \tag{6}$$

where $\omega > 0$ and $\alpha_i, \beta_i \geq 0$, ϵ_t is an uncorrelated random variable, z_t a white noise with variance 1 and σ_t the standard deviation ($\epsilon_t = \sigma_t z_t$). $p \geq 1$ represents the order of dependence of the conditional variance and $q \geq 0$ represents the order of dependence of past shocks. In this work we will study GARCH$(1, 1)$ simulated model.

B. Complex Networks

graphs are a very appropriate modeling tool to represent a set of elements that interact and which exhibit emergent collective properties, in which the elements and their relations are represented by *nodes* (or *vertices*) and *links* (or *edges*), respectively. Such a graph has non-trivial topological properties, due to the specific characteristics of the system it represents [23]. We call this graphical representation of *complex network*. We will use these two terms, graph and network, interchangeably.

1) Graph Terminology and Concepts: A *graph* (G) is then an ordered pair $(V(G), E(G))$, where $V(G)$ represents the set of *nodes* and $E(G)$ the set of *links* between pairs of elements of the set $V(G)$. The number of nodes, also known as the *size* of the graph, is written as $|V(G)|$ and the number of links as $|E(G)|$. A k-graph is a graph of size k.

Two nodes are *neighbors* or *adjacent* if they are connected by a link, that is, if $(v_i, v_j) \in E(G)$ then v_i and v_j are neighbors. We can distinguish between *directed* links, which connect a source node to a target node, and *undirected* links, when there is no such concept of orientation. In the first case the graph is called directed or *digraph*.

A graph can also be *weighted*, this means that at each link (v_i, v_j) is associated with a weight (or cost) $w_{i,j}$, and this weight can be positive or negative.

A graph is classified as *simple* if it does not contain multiple links, two or more links connecting the same pair of nodes, and it does not contain self-loops (a link connecting a node to itself).

a) Path: A *path* is a sequence of nodes in which each consecutive pair of nodes in the sequence is connected by a link.

b) Connectivity: The concept of *connectivity* is extremely important in networks. We say that two nodes are then *connected* if there is a path between them and are *disconnected* if such a path does not exist. In undirected graphs, if nodes v_i and v_j are connected and nodes v_j and v_k are connected, then v_i and v_k are also connected. This property can be used to partition the nodes of a graph in non-overlapping subsets of connected nodes known as *connected components* [6]. Although there is at least one path between any two nodes in the same component, there is no path between nodes belonging to different components [4].

2) Topological Metrics: There is a vast set of topological metrics of available graphs [6], each reflecting some particular features of the system under analysis. In our work we are essentially concerned with studying a specific set of simple metrics. We present below a brief description of them.

a) Average degree: The degree of a graph is a fairly important local property of each node, this represents the number of links that the node has for the other nodes, in undirected graphs [4]. We denote by k_i the degree of the i-th node.

In digraphs, we distinguish between the *in-degree*, k_i^{in}, and the *out-degree*, k_i^{out}. The first represents the number of links that point to node v_i, and the second the number of links that point from node v_i to other nodes. The total degree, k_i, in a digraph is given by the sum of the two.

In weighted graphs we may want to obtain the weighted degree that is similar to the previous measure, the difference is that instead of adding the quantity of connections, we sum the weights of each of the links [6].

From this measure we can obtain the average degree ($\bar{k}$) that can be easily obtained by calculating the arithmetic mean of the degrees of all nodes in the graph.

b) Average path length: A path is a sequence of nodes in which each consecutive pair of nodes in the sequence is connected by a link. It may also be useful to think of the path as the sequence of links that connect those nodes. In digraphs the path follows the direction of the source node for the target node.

We denote by $\bar{d}$, the arithmetic mean of the shortest paths (d) among all pairs of nodes (both ways for directed graphs), the path length being the number of links, or the sum of the links weights if the graph is weighted, in the path [4]. It should be noted that $\bar{d}$ is measured only for the node pairs that are in the same component.

For a directed graph the average path length is given by:

$$\bar{d} = \frac{1}{|V(G)|(|V(G)|-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{|V(G)|} d_{i,j} \tag{7}$$

c) Global clustering coefficient: This measure (C), also called global transitivity, measures the total number of closed triangles in the graph, that is, it measures the degree to which the nodes in a graph tend to cluster. It is calculated by the ratio of the number of closed triangles ($N_{\triangle}$) to the number of possible triangles, that is, the amount of connected triplets of nodes (N_3) [6]. In this work we will call it only as a *clustering coefficient*.

For undirected and unweighted graphs, we mathematically have to:

$$C = \frac{3N_{\triangle}}{N_3} \tag{8}$$

Factor three explains the fact that each triangle can be seen to consist of three different triangles, one with each of the nodes as the central node, and ensures that $0 \leq C \leq 1$. For directed graph the direction of the edges is ignored.

For weighted graphs there are several generalizations of clustering coefficient, here we use the definition by A. Barrat [24], this is a local vertex-level quantity, its formula is:

$$C_i = \frac{1}{k_i^w(k_i-1)} \sum_{j,h} \frac{(w_{i,j}+w_{i,h})}{2} a_{i,j}a_{i,h}a_{j,h} \tag{9}$$

where k_i^w is the weighted degree, $a_{i,j}$ are elements of the adjacency matrix, k_i is the degree and $w_{i,j}$ are the weights.

d) Number of communities: Number of communities(S) measures the number of denser subgraphs in a network, that is, subsets of nodes within the graph such that connections between the nodes are denser than connections to the rest of the network.

The function we used to help us calculate this metric tries to find densely connected subgraphs, also called communities here, via random walks. The idea is that short random walks tend to stay in the same community. This function is the implementation of the Walktrap community finding algorithm [25].

e) Modularity: Measures how good the division of the graph is in specific communities, that is, how different are the different nodes, belonging to different communities, from each other. A high modularity value, (Q), indicates a graph with a dense internal community structure, that is, with many edges between nodes within communities and sparse connections between nodes of different communities [26].

If a particular network is split into c communities, Q can be calculated from the symmetric $c \times c$ mixing matrix $E(G)$ whose elements along the main diagonal, e_{ii}, give the fraction of connections between nodes in the same community i while the other elements, $e_{ij}(i \neq j)$ identify the fraction of connections between nodes in the different communities i and j. The calculation of Q can then be performed as follows [6], [26]:

$$Q = \sum_i [e_{ii} - (\sum_j e_{ij})^2] \tag{10}$$

The situation $Q = 1$ identifies networks formed by disconnected modules.

III. Mappings from Time Series to Complex Networks

Several network-based time series analysis approaches have been recently proposed, based on mapping time series to the network domain. The objective is to map a time series for a complex network using a particular concept, in our particular work, based on the concepts of visibility and probability of transition existing in the literature.

A. Natural Visibility Graph

This method was proposed for the first time by Lacasa et al. [10]. Named natural visibility graph (NVG), each node in the graph corresponds, in the same order, to the time series data and two nodes are connected if there is a line of visibility between the corresponding data points, that is, if it is possible to draw a straight line in the time series that joins the two corresponding data points that intercepts no data "height" between them. If we consider each time instant as a node of a graph, then two nodes are connected if the tops of the corresponding vertical bars are visible to each other, that is, if there is a straight line from the top of the two bars that does not intersect other bars. This mapping is illustrated in figure 1 with a toy time series and the resulting network.

The resulting graph has as many nodes as the number of observations in the time series. The nodes are numbered sequentially in time and each node corresponds to an observation (t_a, y_a). Two nodes (t_a, y_a) and (t_b, y_b) are connected (have visibility) if any other observation (t_c, y_c) with $t_a < t_c < t_b$ satisfies:

$$y_c < y_b + (y_a - y_b)\frac{(t_b - t_c)}{(t_b - t_a)} \tag{11}$$

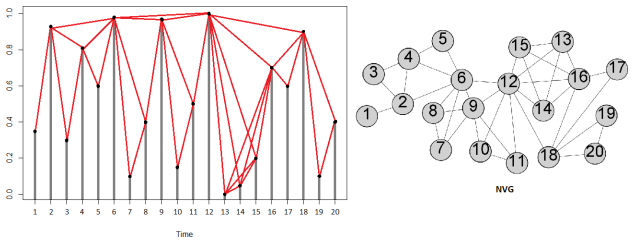

Fig. 1: On the left side, we present the plot of a toy time series and, on the right side, the network generated by the natural visibility algorithm. The red lines in the time series plot represent the lines of visibility (and hence the links of the graph) between all data points.

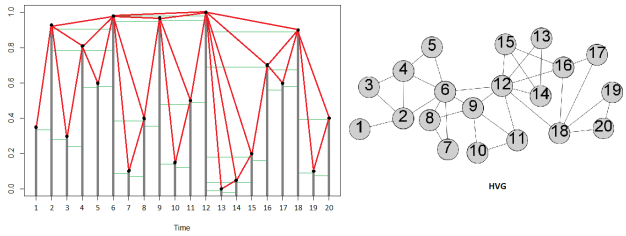

Fig. 2: On the left side, we present the plot of a toy time series and, on the right side, the network generated by the horizontal visibility algorithm. The green lines represent the horizontal lines of visibility between all the data points and the red lines the respective connections between the points.

The graphs obtained always have the following characteristics [10]:

- **Connected:** each node sees at least its nearest neighbors (left-hand side and right-hand side).
- **Undirected:** the way the algorithm is built up, there is no direction defined in the links. However, this direction could be defined considering the direction of the time axis.
- **Invariant under affine transformations of the series data:** the visibility criterion is invariant under rescheduling of both the horizontal and vertical axis, as well as in horizontal and vertical translations.
- **"Lossy":** some information regarding the time series is inevitably lost in the mapping from the fact that the network structure is completely determined in the (binary) adjacency matrix. For instance, two periodic series with the same period as $Y_a = ..., 3, 1, 3, 1, ...$ and $Y_b = ..., 3, 2, 3, 2, ...$ would have the same visibility graph, albeit being quantitatively different. One possible solution would be the use of weighted networks, where weights determine the height difference of the associated data, for example.

B. Horizontal Visibility Graph

In order to reduce the computational complexity associated to NVG, Luque et al. [27] proposed in 2009 a simplified NVG method called the horizontal visibility graph (HVG), which inherits all NVG features mentioned above.

In this alternative, two nodes in the graph are connected if it is possible to draw a horizontal line in the time series joining the two vertical bars, corresponding to the two data, which does not intercept any height of the intermediate data. In the figure 2 we give a simple illustration of this method, with a toy time series and the resulting network.

Formally, two nodes (t_a, y_a) and (t_b, y_b) are connected, have visibility, if the following condition is fulfilled:

$$y_a, y_b > y_c \tag{12}$$

for all t_c such that $t_a < t_c < t_b$.

The HVG is always a subgraph of the NVG associated with the same time series, for example, if we analyze both graphs in the figures 1 and 2 we can easily verify that all the links present in the HVG are present in NVG, but there are links in NVG that are not in HVG, two examples are links $(7, 9)$ and $(13, 15)$. Therefore, the HVG nodes will always have a degree less than or equal to the nodes of the corresponding NVG, since they will have "less visibility" and consequently will have less quantitative information.

C. Quantile Graph

Quantil graphs (QG) were introduced by Campanharo et al. [11]. It is a different approach from previous methods of visibility, but captures oscillations over time. This method divides the time series into Q quantiles, $q_1, q_2, ..., q_Q$, and each quantile, q_i, is associated to a node v_i of the graph. So the graph has as many nodes as the number of quantiles. Two nodes v_a and v_b are connected by a weighted directed link $(v_a, v_b, w_{a,b})$, where the weight $w_{a,b}$ represents the number of times an observation (t_n, y_n), belonging to the quantile q_a, is followed by an observation $(t_{n+1}, y_{n+1}$, belonging to the quantile q_b. The weights are normalized such that the adjacency matrix becomes a Markov transition matrix, where $\sum w_{a,b} = 1$. The resulting networks are weighted and directed. This mapping is illustrated in figure 3 with a toy time series and the resulting network.

These networks have a significant loss of information on small amplitude variations, especially if the value of Q is very small. Its connectivity represents the causal relationships contained in the dynamics of the process it represents.

For this work we chose to use 50 quantiles and let us refer to the generated QGs as 50-QG or simply Q50.

IV. Clustering of Time Series Models

The purpose of cluster analysis is to discover the natural groupings of a set of patterns, points, or objects. It consists of the empirical formation of groups of objects, called clusters, with high intra-cluster similarity and low inter-cluster similarity. That is, given a representation of n objects, the goal is to find k clusters based on a measure of similarity, so that the similarities between objects in the same cluster are high,

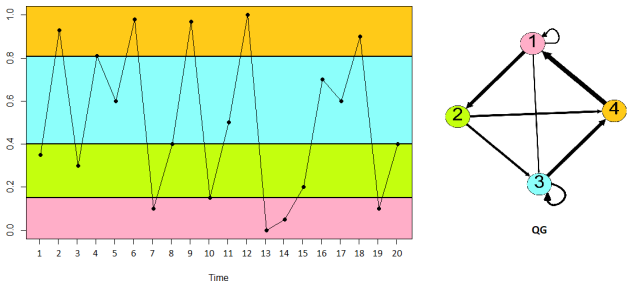

Fig. 3: On the left side, we present the plot of a toy time series and, on the right side, the network generated by the quantile algorithm. The different colors represent the region (in the time series plot) corresponding to the different quantiles (in this case $Q = 4$). Repeated transitions between quantiles result in edges in the network with larger weights represented by thicker lines.

whereas the similarities between objects in different clusters are low [28].

In the clustering of time series, we can distinguish between two main categories, the one that performs clustering on a set of time series with the purpose of grouping them in different clusters, and the one that performs clustering on "windows" of a single time series whose objective is to find similarities and differences between different windows of time. Here we focus on the first category.

One of the major problems in time series clustering analysis is the choice of a relevant metric to perform grouping. Approaches involving the measure of similarity between global characteristics of the time series were proposed to improve old approaches based on similarity measures (eg, Euclidean distance, Dynamic Time Warping, autocorrelation, spectrum, ...), between the actual observations of time series [16]. In this paper we present a new approach in this direction that consists of resorting to the science of complex networks.

More precisely, we propose an approach that involves the measurement of similarity between characteristics, where the set of resources (topological metrics) used for clustering analysis are extracted from complex networks that are created for each of the time series that we want to group. Therefore, the proposed approach involves the following tasks:

1) For each of the time series under analysis, we generated the corresponding natural visibility graph (NVG), the horizontal visibility graph (HVG) and the quantile graph (QG). For the QG we use a total of 50 quantiles.
2) For each of the networks (or graphs) we have computed the five topological metrics mentioned in subsection II-B: $\bar{k}$, $\bar{d}$, S, C, and Q.
3) Since the interval of each of the calculated measures can vary significantly, we apply the Min-Max normalization so that each measure is in the range $[0, 1]$, preventing some measures from dominating the others in the grouping process.
4) After calculating all the metrics, we obtain a vector with fifteen topological features, which will be scaled through the principal component analysis (PCA) [29] and then by the t-distributed stochastic neighbor embedding (t-SNE) technique [30], that will feed the k-means.

All the computations are performed in `R` [31] (version 3.4.4), using specific packages, such as `igraph` [32] for graph generation and calculation of metrics, and `timeSeries` [33], `fracdiff` [34], `fGarch` [35], and `rugarch` [36], for the simulation of some time series models mentioned below. The simulation of some time series models required the implementation of the appropriate procedure.

The main idea of this new approach is to show that the use of complex networks can easily distinguish time series models from a wide set of different models, showing that the topological characteristics of the networks corresponding to the series can capture the global nature of the same. We also want to show that the joining of the visibility methods and the quantile methods improves the clustering of the series in contrast to the use of only one mapping method. This is our main contribution, since until now no other paper joins different concepts of network mapping.

A. Dataset

In order to apply the proposed approach we decided to simulate a large set of time series models that are most common and widely used in the statistical theory and practice of time series analysis. We generate, using `R` software and appropriate packages, 100 sample of size $T = 10000$ of each of the 10 models previously presented (subsection II-A), in a total of 1000 time series.

We refer to these models as follows:

- **White noise:** only `White Noise`;
- **AR models:** `AR(1)-0.5`, `AR(1)0.5` for AR(1) processes with parameters $\phi_1 \in \{-0.5, 0.5\}$; and `AR(2)` for AR(2) process;
- **ARIMA models:** only `ARIMA(1,1,0)`;
- **ARFIMA models:** `ARFIMA(1,0.4,0)-0.5`, `ARFIMA(1,0.4,0)0.5` for ARFIMA$(1, 0.4, 0)$ processes with parameters $\phi_1 \in \{-0.5, 0.5\}$;
- **SETAR models:** only `SETAR(1)`;
- **INAR models:** only `INAR(1)`;
- **GARCH models:** only `GARCH(1,1)`.

The time series are then mapped into networks using the NVG, HVG and 50-QG methods. The resulting 3000 ($1000*3$) networks are characterized by the topological metrics. We obtain a data frame of 15 variables (features) and 1000 instances.

B. Results

We performed 7 types of clustering analysis, using 7 different feature vectors, namely, the metrics of only one of the mapping methods (3 different vectors), two to two metrics of the mapping methods (3 different vectors), and finally, a vector containing all the metrics obtained from the three different mapping methods, and we compared the results with the true classification, (the original time series models), using

Mappings	Adjusted Rand Index	Average Silhouette
NVG	0.36	0.51
HVG	0.63	0.66
Q50	**0.64**	**0.73**
NVG-HVG	0.68	0.63
NVG-Q50	**0.78**	**0.75**
HVG-Q50	**0.79**	**0.73**
NVG-HVG-Q50	**0.80**	**0.73**

TABLE I: Clustering evaluation metrics for the different clustering analysis. The results refer to the evaluation metrics for the dataset.

the clustering evaluation metrics mentioned earlier, in order to support our assertion that using two different concepts of mapping methods is an advantage for better recognition of network characteristics (and time series inevitably).

We divide this results into two parts: the results obtained from the principal components analysis (as a dimensionality reduction technique) and then the results obtained from the clustering analysis (using k-means algorithm and knowing *a priori* the correct number of clusters in the dataset, that is, $k = 10$).

To evaluate the results of clustering, we chose two evaluation metrics, namely, *adjusted Rand index* [37] and *average silhouette*. The two measures have different functions, the first is a measurement of the accuracy of the results: compares the clusters obtained with the true clusters. The second measures the quality of clusters obtained without knowledge of the true clusters. Adjusted Rand index takes values between -1 and 1. It is negative if the index is less than the expected index. Its expected value is 0 in the case of random clusters. A larger Adjusted Rand Index means a higher agreement between two partitions. And the average silhouette takes values between -1 and 1, where a high value indicates that the object is well compatible with its own cluster but not with neighbor clusters. If most objects have a high value, the cluster configuration is appropriate.

The results of the clustering evaluation metrics we choose (adjusted Rand index and average silhouette) obtained for the different combinations of feature vectors are presented in the table I, columns 2 and 3, respectively.

The colors represent the two maximum values of the corresponding column, with the darker color highlighting the maximum value and the lighter color the second maximum value.

We can observe that the feature vector that is closest to the real clusters is the one corresponding to the junction of the three proposed mapping methods, (NVG, HVG and 100-QG), with a value of 0.80 in a range of $[-1, 1]$. The vectors that obtained the best values (0.75 and 0.73, in a range of $[-1, 1]$) of the average silhouette are the vectors corresponding to the metrics of the graphs obtained by the two concepts of mapping together. We thus show that the best results are those that use in the dataset the two types of time series mapping concepts in networks (concept of visibility and concept of probabilities of transition). Thus, we prove that the addition of more information about the data do translate into a better result, as we expected. This evidence is further reinforced by the fact that the combination of the two visibility methods yields (NVG and HVG) 0.68 for adjusted Rand index, which is much lower than the best results obtained (0.80, 0.79 and 0.78) that correspond to the vectors obtained from the visibility and quantile methods.

If we focus on feature vector corresponding to just one type of mapping, we note that the that best capture the characteristics of the time series are the 50-QGs with an adjusted Rand index of 0.64 compared to the NVG and HVG that obtained 0.36 and 0.63, respectively, and the average silhouette value is the highest (0.64). This is in agreement with the expected one, since this method better captures the variability of the observations of the time series. And the visibility of methods capture more global structural properties of time series.

Let us now analyze in more detail some results obtained. First we will analyze the results obtained by the PCA corresponding to the characteristic vector using the three mappings (the best value of adjusted Rand index). And then we will analyze the results of clusters obtained for this vector.

1) PCA Results: Figure 4 represent the biplot obtained by the PCA for the feature vector that obtained the best adjusted Rand index.

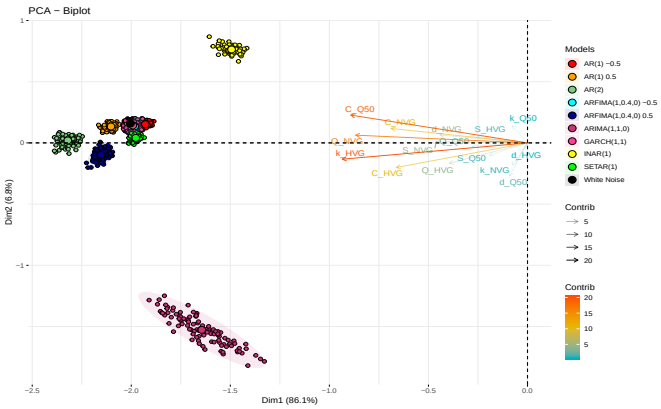

Fig. 4: Results of the PCA analysis. Objects belonging to different groups have different colors, and the arrows represent the contributions of the features to the PCs (the larger the size, sharpness, and closer to orange the greater the contribution of the feature).

We can start by noting that several of the objects belonging to different classes are actually separated into clusters in this bidimensional space, showing that most of the networks are first grouped correctly using the two principal components.

We can also verify that the apparently more dissimilar objects are those corresponding to models with specific characteristics such as trend (`ARIMA(1,1,0)`), periodicity (`AR(2)` and `ARFIMA(1,0.4,0)0.5`), counting (`INAR(1)`), and regime changes (`SETAR(1)`). On the other hand there is a greater difficulty in distinguishing the networks corresponding to the `ARFIMA(1,0.4,0)-0.5`, `White Noise` and

`GARCH(1,1)` models, which are processes that are somewhat similar, mainly the last two, and therefore expected.

We can still verify that, just as the arrows in the plot themselves suggest, different topological metrics of the three mapping methods contribute to distinguish different time series models. The $\bar{d}$ and C of Q50, $\bar{k}$ of NVG, and C of NVG and HVG, contribute to distinguish second PC, that is, `INAR(1)`, `ARIMA(1,1,0)`, `AR(2)`, and `ARFIMA(1,0.4,0)0.5` models. The C and Q of NVG, $\bar{k}$ and C of HVG, and C of 50-QG, contribute to distinguish first PC, that is, `AR(2)`, `ARFIMA(1,0.4,0)0.5`, `AR(1)0.5`, `ARIMA(1,1)` and `INAR(1)`.

2) Custer Results: In figure 5 we present an enhanced jitter strip chart, where the width of the jitter is controlled by the density distribution of the data within each class. We can see an almost perfect attribution of the objects by the different clusters, with the exception are `ARFIMA(1,0.4,0)-0.5`, `GARCH(1,1)` and `White Noise` networks which are not distinguishable and are assigned to the same clusters. We conclude that these are the most similar models from the point of view of the complex networks, and consequently more difficult to distinguish, that in the perspective of the analysis of time series have very similar characteristics.

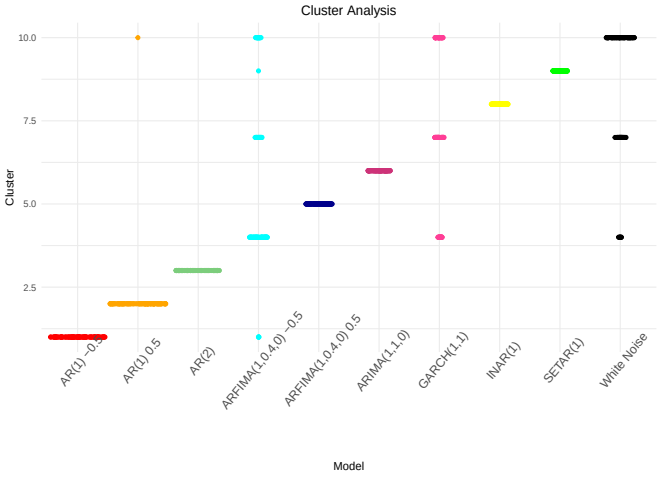

Fig. 5: Plot the distribution of the objects corresponding to the time series models by the different clusters.

As we specify *a priori* in the algorithm k-means that the number of clusters is 11, the algorithm "divides" these three models into three different clusters. But we can see from the breadth of the traces that this distribution is not uniform, this emphasizes a possible "capacity" to distinguish, mainly, the `ARFIMA(1,0.4,0)-0.5` and `White Noise` models, which are mostly distributed by cluster 4 and 10, respectively.

Although there is not a completely perfect assignment by clusters, we can conclude that it is a good and relevant result.

V. Conclusion

Classical approaches to time series analysis present severe limitations when analyzing sets of time series. A recent and very promising conceptual approach relies on mapping the time series to complex networks, where the large set of network science methodologies can help in grouping time series.

Our objective with this work is to contribute to the improvement of the methods of clustering of time series, using a complementary area. For this we construct a dataset of 3000 synthetic complex networks, distributed by 10 types of time series models, and we analyse using data mining tools.

The results show that our approach is able to group almost all different time series models using a set of basic topological metrics of complex networks based on different mapping methods.

The main advantage of the proposed approach is to be a completely nonparametric method that can serve as a solution to the parametric and statistical methods of time series analysis. We show that different mappings complement each other, identifying different characteristics of time series. Results show the validity and discrimination power, we were able to distinguish networks corresponding to non-stationary from stationary time series models, counting from non-counting time series models, periodic from non-periodic time series models, and state models from state models time series. More specifically, out of 10 different network types we can distinguish perfectly 7 from them. However, we could not group the networks corresponding to the `ARFIMA(1,0.4,0)-0.5`, `GARCH(1,1)` and `White Noise` models into different clusters, given their very similar characteristics.

In future work, we want to explore new sets of topological metrics as well as new types of mapping of series for network, in order to improve our results, more specifically, to be able to separate perfectly the models that we can not achieve in this work.

References

[1] Box GE, Jenkins GM, Reinsel GC, Ljung GM. Time series analysis: forecasting and control. John Wiley & Sons; 2015.
[2] Paparrizos J, Gravano L. Fast and accurate time-series clustering. ACM Transactions on Database Systems (TODS). 2017;42(2):8.
[3] Newman ME. The structure and function of complex networks. SIAM review. 2003;45(2):167–256.
[4] Barabási AL. Network Science. Cambridge University Press; 2016.
[5] Kantarci B, Labatut V. Classification of complex networks based on topological properties. In: Cloud and Green Computing (CGC), 2013 Third International Conference on. IEEE; 2013. p. 297–304.
[6] Costa LdF, Rodrigues FA, Travieso G, Villas Boas PR. Characterization of complex networks: A survey of measurements. Advances in physics. 2007;56(1):167–242.
[7] Zhang J, Small M. Complex network from pseudoperiodic time series: Topology versus dynamics. Physical review letters. 2006;96(23):238701.
[8] Gao Z, Jin N. Complex network from time series based on phase space reconstruction. Chaos: An Interdisciplinary Journal of Nonlinear Science. 2009;19(3):033137.
[9] Marwan N, Donges JF, Zou Y, Donner RV, Kurths J. Complex network approach for recurrence analysis of time series. Physics Letters A. 2009;373(46):4246–4254.
[10] Lacasa L, Luque B, Ballesteros F, Luque J, Nuno JC. From time series to complex networks: The visibility graph. Proceedings of the National Academy of Sciences. 2008;105(13):4972–4975.
[11] Campanharo AS, Sirer MI, Malmgren RD, Ramos FM, Amaral LAN. Duality between time series and networks. PloS one. 2011;6(8):e23378.

[12] Zhu G, Li Y, Wen PP. Analysis and classification of sleep stages based on difference visibility graphs from a single-channel EEG signal. IEEE journal of biomedical and health informatics. 2014;18(6):1813–1821.
[13] Shao ZG. Network analysis of human heartbeat dynamics. Applied Physics Letters. 2010;96(7):073703.
[14] Campanharo A, Ramos F. Distinguishing different dynamics in electroencephalographic time series through a complex network approach. Proceeding Series of the Brazilian Society of Computational and Applied Mathematics. 2017;5(1).
[15] Telesca L, Lovallo M. Analysis of seismic sequences by using the method of visibility graph. EPL (Europhysics Letters). 2012;97(5):50002.
[16] Hennig C, Meila M, Murtagh F, Rocci R. Handbook of cluster analysis. CRC Press; 2015.
[17] Shumway RH, Stoffer DS. Time series analysis and its applications. Springer; 2017.
[18] Cryer JD, Chan KS. Time Series Analysis With Applications in R. New York: Springer; 2008.
[19] Franses PH, Van Dijk D. Non-linear time series models in empirical finance. Cambridge University Press; 2000.
[20] Tong H. Threshold models in time series analysis—30 years on. Statistics and its Interface. 2011;4(2):107–118.
[21] Silva I, Silva ME, Pereira I, Silva N. Replicated INAR(1) processes. Methodology and Computing in applied Probability. 2005;7(4):517–542.
[22] Tsay RS. Analysis of financial time series. vol. 543. John Wiley & Sons; 2005.
[23] Albert R, Barabási AL. Statistical mechanics of complex networks. Reviews of modern physics. 2002;74(1):47.
[24] Barrat A, Barthelemy M, Vespignani A. The Architecture of Complex Weighted Networks: Measurements and Models. In: Large Scale Structure And Dynamics Of Complex Networks: From Information Technology to Finance and Natural Science. World Scientific; 2007. p. 67–92.
[25] Pons P, Latapy M. Computing communities in large networks using random walks. In: International symposium on computer and information sciences. Springer; 2005. p. 284–293.
[26] Clauset A, Newman ME, Moore C. Finding community structure in very large networks. Physical review E. 2004;70(6):066111.
[27] Luque B, Lacasa L, Ballesteros F, Luque J. Horizontal visibility graphs: Exact results for random time series. Physical Review E. 2009;80(4):046103.
[28] Jain AK. Data clustering: 50 years beyond K-means. Pattern recognition letters. 2010;31(8):651–666.
[29] Abdi H, Williams LJ. Principal component analysis. Wiley interdisciplinary reviews: computational statistics. 2010;2(4):433–459.
[30] Van Der Maaten L. Accelerating t-SNE using tree-based algorithms. Journal of machine learning research. 2014;15(1):3221–3245.
[31] R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria; 2018. Available from: https://www.R-project.org.
[32] Csardi G, Nepusz T. The igraph software package for complex network research. InterJournal. 2006;Complex Systems:1695. Available from: http://igraph.org.
[33] Wuertz D, Setz T, Chalabi Y. timeSeries: Rmetrics - Financial Time Series Objects; 2017. R package version 3042.102. Available from: https://CRAN.R-project.org/package=timeSeries.
[34] original by Chris Fraley S, U Washington, port by Fritz Leisch at TU Wien; since 2003-12: Martin Maechler; fdGPH SR, fdSperio, etc by Valderio Reisen, Lemonte A. fracdiff: Fractionally differenced ARIMA aka ARFIMA(p,d,q) models; 2012. R package version 1.4-2. Available from: https://CRAN.R-project.org/package=fracdiff.
[35] Wuertz D, Setz T, Chalabi Y, Boudt C, Chausse P, Miklovac M. fGarch: Rmetrics - Autoregressive Conditional Heteroskedastic Modelling; 2017. R package version 3042.83. Available from: https://CRAN.R-project.org/package=fGarch.
[36] Ghalanos A. rugarch: Univariate GARCH models.; 2018. R package version 1.4-0.
[37] Campello RJ. A fuzzy extension of the Rand index and other related indexes for clustering and classification assessment. Pattern Recognition Letters. 2007;28(7):833–841.