Common Dissimilarity Measures are Inappropriate for Time Series Clustering

Cássio M. M. Pereira, Rodrigo F. de Mello
Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação
São Carlos, SP
{cpereira,mello}@icmc.usp.br

Abstract: Clustering algorithms have been actively used to identify similar time series, providing a better understanding of data. However, common clustering dissimilarity measures disregard time series correlations, yielding poor results. In this paper, we introduce a dissimilarity measure based on series partial autocorrelations. Experiments compare hierarchical clustering algorithms using the common dissimilarity measures, such as Euclidean Distance and Dynamic Time Warping, to cluster time series following Box-Jenkins Auto-Regressive models. Results show that our dissimilarity measure produces better results for both synthetic and real data sets in terms of the Adjusted Rand Index and Normalized Hubert Γ statistic. Our findings confirm that the choice of dissimilarity measure is crucial for improving time series clustering quality.

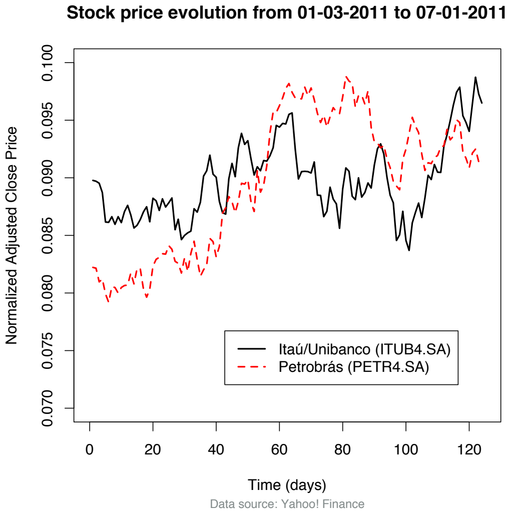
1 Introduction

Clustering [1, 2, 3] is one of the most employed Machine Learning techniques [4], since it does not require labeled data. For real-world problems, where human labeling is too costly, or there is simply no supervision available, clustering is a specially useful tool.

Consider, for example, a stock market investor who wishes to find the best stocks to create a portfolio. Ideally, the portfolio should be diversified, containing several companies from different economy sectors. Diversification allows the losses in one sector to be compensated by the profits in another.

Figure 1 illustrates the benefits of diversification by showing the stock price evolution of two Brazilian companies, one from the oil sector (Petrobrás) and the other from banking (Itaú/Unibanco). In several moments, the lows in one stock were compensated by the highs of the other. Time series clustering provides an intuitive understanding of the relations among stock price series. An investor can choose one or several companies from each cluster, better diversifying his/her portfolio and minimizing the loss margin.

Clustering algorithms can identify which time series are more similar, providing a



Data source: Yahoo! Finance

Figure 1: Daily close prices of two Brazilian companies, one from the oil market sector (Petrobrás) and one from banking (Itaú/Unibanco). The moments when one of the stocks was low were compensated by the other highs. Clustering of stock price series may help an investor diversify his/her portfolio, minimizing the margin for loss.

better understanding of data. Formally, a clustering algorithm can be seen as a function f which maps every object $x_i \in \mathcal{X}$ to a cluster $C_j \subsetneq \mathbf{C}$, i.e., $f(x_i) \mapsto C_j$.

Several clustering algorithms are available, which can be generally divided into: 1) partitional, which generate a set (partition) of disjoint subsets (clusters) containing objects; and 2) hierarchical, which output a sequence of nested partitions.

Hierarchical algorithms present interesting advantages over the others. First, they do not require a pre-defined number of clusters, which is usually an input to partitional methods. Second, the output of a hierarchical algorithm can be represented by a dendrogram, an intuitive visualization method that simplifies interpretation by domain specialists. Third, hierarchical methods can work without data objects, solely requiring a proximity matrix. This feature is specially interesting when dealing with private information, such as banking data bases.

In this paper, we show how traditional dissimilarity measures disregard the most important time series characteristic, i.e., the serial correlation. To overcome that drawback, we propose a dissimilarity measure based on series partial autocorrelations.

Experiments evaluated traditional clustering algorithms, namely Single-linkage, Complete-linkage and Average-linkage [1], along with various dissimilarity measures, such as Euclidean Distance, Dynamic Time Warping [5] and correlation-based measures, to cluster time series data bases.

Several synthetic data bases were generated in order to have known clusters as a golden truth for evaluation purposes. The clusters corresponded to series modeled by Auto-Regressive (AR) processes [6]. Next, experiments were conducted with real time series, originated from physiological sensor measurements.

This paper is organized as follows. In the next section, the basic concepts of hierarchical clustering are presented. Section 3 contains a brief introduction of time series and the AR models proposed by Box-Jenkins. Next, Section 4 introduces related work. In Section 5, the experimental methodology is discussed, while results are presented in Section 6. Lastly, Sections 7 and 8 contain the discussion and conclusion.

2 Hierarchical Clustering

According to Jain and Dubes [1], clustering is a partitioning of a finite set of objects. A proximity matrix, whose rows and columns represent objects, defines their relations. If objects are, for example, points in a multidimensional space, then the proximity among them can be given by a distance function, such as the Euclidean distance.

Formally, a hierarchical clustering method can be defined as a procedure to transform a proximity matrix into a sequence of nested partitions [1]. For example, let $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ be a set of objects. A partition $\mathbf{C}$ divides $\mathcal{X}$ into subsets $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$ that satisfy: $C_i \cap C_j = \emptyset (\forall i \neq j)$ and $C_1 \cup C_2 \cup \ldots \cup C_k = \mathcal{X}$. A partition $\mathbf{C_1}$ is nested into $\mathbf{C_2}$ if every component of $\mathbf{C_1}$ is a subset of a component of $\mathbf{C_2}$.

Hierarchical algorithms can be either agglomerative or divisive [1]. All agglomerative algorithms choose, based on the proximity matrix, the most similar groups to be merged. The difference among the various algorithms is in how they update the proximity matrix for the next iteration, assigning a new distance to the groups after merging them. In the Single-linkage algorithm, the proximity between groups is given by the minimum distance between objects within different clusters. Conversely, for Complete-linkage the distance is given by the maximum dissimilarity between objects. Finally, for Average-linkage, the distance is the average among all pairs of objects belonging to clusters.

Each update criteria has a bias towards a type of hierarchy, which has pros and cons. The advantage of Single-linkage is its ability to find arbitrarily shaped clusters, but it has the drawback of being highly sensitive to noise and outliers [1]. Complete-linkage, on the other hand, is less susceptible to noise and outliers, but tends to break large clusters. It also cannot deal with arbitrarily shaped clusters as Single-linkage [1]. Lastly, Average-linkage represents a compromise between Single and Complete-linkage, by being less susceptible to noise and outliers, but unable to find arbitrarily shaped clusters [7].

After obtaining the hierarchy, it is usually desired to select a partition at some level, where the number of clusters fits a desired value. To do that, a cut is made in the dendrogram. Figure 2 presents a sample dendrogram with a horizontal cut to obtain four clusters.

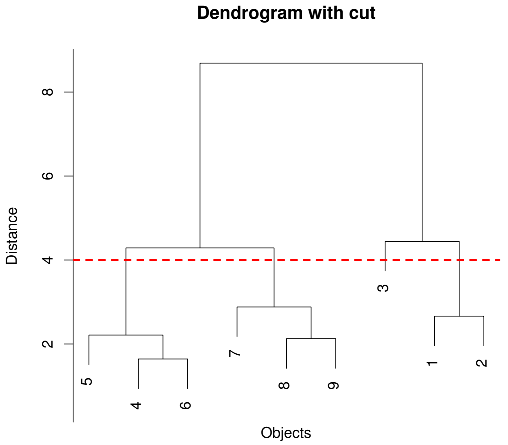


Figure 2: Sample cut into four clusters in a dendrogram obtained using Complete-linkage over a data set with nine objects. The resulting partition is: $\mathbf{C} = \{\{x_1, x_2\}, \{x_3\}, \{x_4, x_5, x_6\}, \{x_7, x_8, x_9\}\}$.

3 Time series

Formally, a time series is defined as a sequence $x = (x_1, x_2, \ldots, x_n)$, where n is the number of observations and x_t, $t \in \{1, 2, \ldots, n\}$, is the observed value at time t. For example, a time series can be the precipitation in millimeters in a city, stock market prices, annual number of solar flares, etc.

A common way to analyze time series is by using the parametric models proposed by Box-Jenkins [6, 8]. For example, an Auto-Regressive (AR) model takes current values to be a linear combination of past ones plus white noise. Moving Average (MA) models consider linear combinations of white noise inputs. Auto-Regressive Moving Average (ARMA) combines both models. Lastly, Auto-Regressive Integrated Moving Average (ARIMA) additionally considers non-stationary behaviors. In the experiments reported here, only AR models were used, as they are one of the simplest, but useful to represent several real phenomena [6].

Auto-Regressive models are part of a set of linear stationary models, which make the assumption a series is generated by a linear system, having white noise as input (Figure 3).

Discrete white noise is defined as $\{\varepsilon_t, t \in Z\}$, in which Z is a stochastic process, stationary if $\forall t, E(\varepsilon_t) = \mu_\varepsilon$ and $Var(\varepsilon_t) = \sigma^2_\varepsilon$, where $E(\cdot)$ is the expected value of the random variable, $Var(\cdot)$ is its variance, μ is its mean and σ is the standard deviation. The system, or linear filter, is defined according to Equation 1, with output Z_t, in which a_t is the input, $\Psi(B)$ is the transference function and B is the backward shift operator (defined in Equation 2). That operator retrieves series values at past time instants, such that t is the current time and m is the delay.

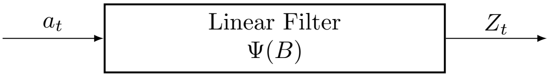


Figure 3: Representation of a linear filter. The input a_t is white noise, which is submitted to a transference function $\Psi(B)$, with output Z_t.

$$Z_t = \mu + a_t + \psi_1 a_{t-1} + \ldots = \mu + \psi(B)a_t \tag{1}$$

$$BZ_t = Z_{t-1}, B^m Z_t = Z_{t-m} \tag{2}$$

In that system, μ defines the level of the series. In case the sequence of coefficients $\psi_j, j \geq 1$ is finite or infinite but convergent, the filter is stable, Z_t is stationary and μ is the mean of the process. Defining $\tilde{Z}_t = Z_t - \mu$, then $\tilde{Z}_t = \psi(B)a_t$. It is possible to rewrite $\tilde{Z}_t$ as a weighted sum of past values plus noise a_t:

$$\tilde{Z}_t = \pi_1 \tilde{Z}_{t-1} + \pi_2 \tilde{Z}_{t-2} + \ldots + a_t \tag{3}$$

In case $\pi_j = 0$ and $j > p$, an auto-regressive model of order p is obtained, commonly denoted as $AR(p)$. Renaming the coefficients π_j to ϕ_j:

$$\tilde{Z}_t = \phi_1 \tilde{Z}_{t-1} + \phi_2 \tilde{Z}_{t-2} + \ldots + \phi_p \tilde{Z}_{t-p} + a_t \tag{4}$$

In the simplest case an auto-regressive model of order $p = 1$, or $AR(1)$, is obtained, which is defined as:

$$\tilde{Z}_t = \phi \tilde{Z}_{t-1} + a_t \tag{5}$$

In that case, Z_t only depends on the value of the previous instant Z_{t-1} and the noise at time t. A useful method to analyze the relationship among observations of a time series is through autocorrelation, which measures the similarity of observations over time. Considering Z_t is a second-order stationary process, then the mean μ and variance σ^2 are independent of time and the autocorrelation depends only on the time delay between observations, but not their absolute positions in time. That allows the autocorrelation function to be expressed as in Equation 6, in which τ is the lag (delay) in time, μ is the mean and σ^2 is the variance.

$$R(\tau) = \frac{E[(Z_t - \mu)(Z_{t+\tau} - \mu)]}{\sigma^2} \tag{6}$$

Besides autocorrelation, another useful measure is the partial autocorrelation, which is simply the autocorrelation between Z_t and $Z_{t-\tau}$ which excludes lags 1 to $\tau - 1$. That simplifies the identification of order p in an Auto-Regressive model.

In this paper, we show how the traditional dissimilarity measures disregard the inherent serial correlations present in time series and how that critically affects clustering performance. A new measure based on series partial autocorrelations is then developed to overcome this drawback (Section 5.2.1).

4 Related work

Classical works on time series similarity [9, 10, 11] have focused on the concept of longest common subsequence. While those techniques may work for certain kinds of time series, namely ones that are similarly shaped, they do not consider the intrinsic correlations present on the time series, which originate from their underlying mathematical model, thus disregarding important information.

Ding et al. [12] conducted a large scale study comparing nine dissimilarity measures for time series, including the L-norms [13]: L_1 norm (Manhattan distance), L_2 norm (Euclidean distance), L_∞; DISSIM [14], which computes the similarity of time series with different sampling rates; Dynamic Time Warping (DTW) [5]; Edit distance based measures [15, 16, 17, 18]: Longest Common Subsequence (LCSS), Edit Sequence on Real Sequence (EDR), Swale, and Edit Distance with Real Penalty (ERP); the Threshold query based similarity search (TQuEST) [19] and Spatial Assembling Distance (SpADe) [20]. The authors concluded that the accuracy of elastic measures such as DTW and Edit distance-based ones converge to that of Euclidean distance as the data set size increases. The accuracy of the Edit-distance based similarities are very close to that of DTW. TQuEST and SpADE are, in general, inferior to DTW. In summary, the authors find that DTW is in general the best similarity measure for mining time series data bases. In this paper, we show that even for simple models, such as the Autoregressive models proposed by Box-Jenkins, DTW cannot find a clustering structure.

In a recent study, Rakthanmanon et al. [21] proposed a method to search and mine trillions of time series subsequences by using Dynamic Time Warping. The authors point out that most time series mining algorithms make extensive use of similarity comparisons and that there is increasing evidence that the classic Dynamic Time Warping (DTW) distance measure is the best one for dissimilarity calculations. While we agree that for various domains DTW has very good results, we show in this paper that even for very simple, yet realistic mathematical time series models, DTW has very poor performance due to the fact that it regards observations of a time series as independent variables.

5 Experimental Methodology

Experiments aimed to show how time series serial correlations affect clustering quality. They also show the poor results obtained with traditional dissimilarity measures that disregard such correlations.

To execute the experiments in a controlled scenario, where clusters were previously known, synthetic time series following Box-Jenkins AR processes were generated. In total, 30 data bases composed of 15 series each (5 per cluster) were used. Each cluster was composed of series following the same AR model, for example, the first cluster was composed of $AR(1)$ series, the second of $AR(2)$ and so forth.

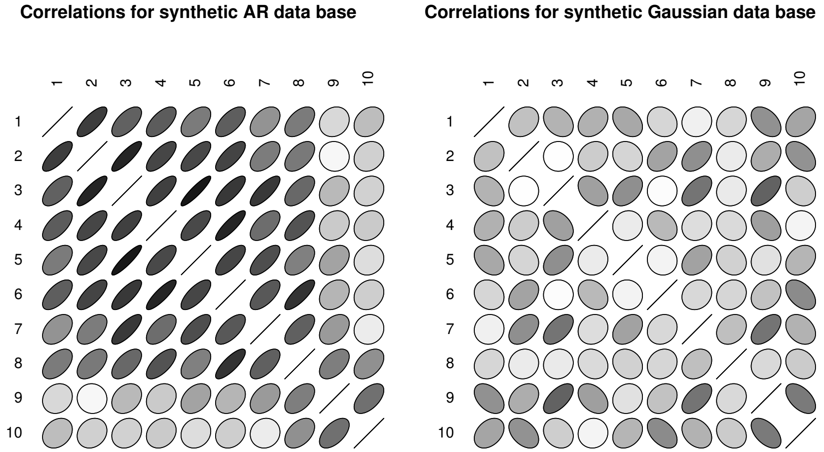
To compare the results of the algorithms on the AR series, we also generated 30 synthetic data bases following Gaussian distributions, i.e., independent and identically distributed data. That allowed the evaluation of clustering quality in the absence of temporal dependencies. Figures 4a and 4b present the correlations among the first ten observations of an $AR(p)$ and a Gaussian data base respectively. One observes there is stronger correlation in the $AR(p)$ base.

Finally, to evaluate the algorithms in a real scenario, a data set from physiological sensors of the ICML 2004 challenge was used[1]. The data set was gathered from wearable sensors. In total, nine sensors were used in different persons of both genders during the practice of physical activities. Table 1 presents a brief description of each sensor functionality.

Thirty physical exercise sessions of different people were used. The time series obtained from sensor measurements of one person during a physical session formed each one of the data bases.

Table 2 summarizes the description of all data bases used in the experiments. Since for ICML bases the clusters were not previously known, the fields “Cluster”, “Model” and “#Series” have missing values.

[1] Data available at: `http://www.cs.utexas.edu/~sherstov/pdmc/`. Last visit: 22 June 2011



(a) Correlations among the observations of an $AR(p)$ data base.

(b) Correlations among the observations of a Gaussian data base.

Figure 4: The charts present the correlations among the first ten observations of an $AR(p)$ and a Gaussian data base. The more the figures resemble a compressed ellipse, the stronger the correlation between observations. Figures closer to a circle indicate absence of correlation. Right tilted ellipses represent positive correlation, while left tilted negative correlations. Darker gray tones also indicate stronger correlation (either positive or negative).

Table 1: Description of sensors used to obtain the time series from the ICML 2004 data base.

# Sensor	Measurement
1	Galvanic skin response
2	Heat flux
3	Approximate body temperature
4	Pedometer
5	Mean skin temperature
6	Longitudinal accelerometer SAD
7	Mean longitudinal accelerometer
8	Transverse accelerometer SAD
9	Mean transverse accelerometer

Table 2: Description of the data bases.

Data	# Bases	# Series per base	# Observ. per series	Type	Cluster	Model	# Series
$AR(p)$	30	15	100	Real values	$AR(1)$	$AR(\phi_1 = 0.5)$	5
					$AR(2)$	$AR(\phi_1 = 0.5, \phi_2 = 0.3)$	5
					$AR(3)$	$AR(\phi_1 = 0.5, \phi_2 = 0.3, \phi_3 = 0.1)$	5
Gaussian	30	15	100	Real values	Gaussian1	$N(\mu = 1.0, \sigma = 0.8)$	5
					Gaussian2	$N(\mu = 1.5, \sigma = 0.8)$	5
					Gaussian3	$N(\mu = 2.0, \sigma = 0.8)$	5
ICML 2004	30	9	765	Real values	?	?	?

The Adjusted Rand Index (ARI) criteria was used to evaluate the algorithms on the $AR(p)$ and Gaussian data bases, since the correct partitions were previously known. ARI is based on the counting of point pairs on which both partitions agree or disagree [22]. Any pair of points can be placed in one out of four categories: (1) they are in the same cluster in both partitions (N_{11}), (2) in different clusters in both partitions (N_{00}), (3) they are in the same cluster in the first partition and in different clusters in the second one (N_{10}) and (4) they are in different clusters in the first partition and on the same in the second (N_{01}). The values N_{11} and N_{00} can be seen as measures of agreement between partitions, while N_{10} and N_{01} as measures of disagreement.

Given two partitions of a set S of N objects, let $U = \{U_1, U_2, \ldots, U_R\}$ be the first partition with R clusters and $V = \{V_1, V_2, \ldots, V_T\}$ the second partition with T clusters. Let $\cap_{i=1}^{R} U_i = \cap_{j=1}^{T} V_j = \emptyset, \cup_{i=1}^{R} U_i = \cup_{j=1}^{T} V_j = S$. It is possible to create a contingency matrix $N_{R \times T}$, whose n_{ij} element is the number of objects in common to clusters U_i and V_j. The Adjusted Rand Index, assuming a hypergeometric distribution as the model of randomness, takes values close to zero for two random partitions and close to one when both partitions are practically identical. The index can be computed according to Equation 7, in which n_{ij} is an element of the contingency matrix, a_i is the sum of the i-th row and b_j is the sum of the j-th column.

$$\text{ARI} = \frac{\sum_{ij} \binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}] / \binom{N}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}] / \binom{N}{2}} \tag{7}$$

For evaluating ICML data bases, the quality index used was the Normalized Hubert Γ statistic between the cophenetic matrix[2], obtained from the clustering procedure, and the proximity matrix. The statistic computes the correlation between the matrices (Equation 8). In that equation, Λ is the cophenetic matrix, D is the original dissimilarity matrix, μ is the

[2] The $\Lambda_{n \times n}$ cophenetic matrix of a set of n objects has as its λ_{ij} element the dissimilarity with which the (i, j) pair of objects were joined.

mean, n is the number of objects and $M = n(n-1)/2$ is the number of pairs of objects. The calculation is done only for the upper triangulars, since both matrices are symmetric. Values close to 1 indicate better compatibility between the clustering and the proximity matrix P.

$$\begin{aligned}\Gamma = &\frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left[\Lambda(i,j)D(i,j)\right] - \mu_\Lambda\mu_d \\ &\cdot \frac{1}{\sqrt{\frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left[\Lambda(i,j)\right]^2 - \mu_\Lambda}} \\ &\cdot \frac{1}{\sqrt{\frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left[D(i,j)\right]^2 - \mu_d}}\end{aligned} \tag{8}$$

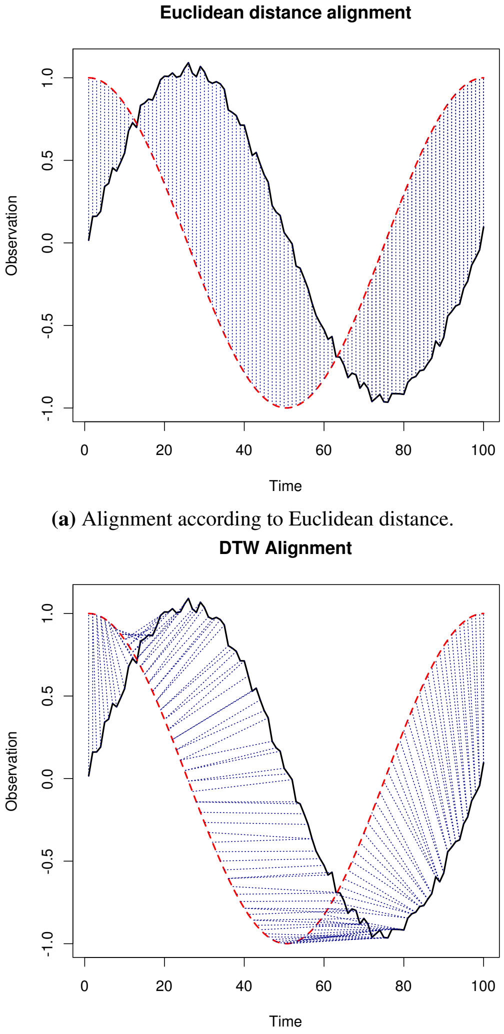
5.1 Dissimilarity Measures

Each one of the clustering algorithms has been executed with at least five dissimilarity measures. Euclidean distance was one of the measures used, although popular, it is not considered to be a robust measure for time series clustering. Dynamic Time Warping was also used, since it is regarded as one of the best time series distance functions. Two correlation-based measures were also used, Pearson and Spearman, but transformed into dissimilarities since they originally compute similarity. Lastly, a distance based on the Cosine similarity was used.

5.1.1 Euclidean Distance Euclidean distance is one of the most popular dissimilarity measures in Machine Learning. It is a specific case of the Minkowski norm (Equation 9) with $p = 2$. Euclidean distance favors the discovery of globular clusters. For time series it is not an ideal distance, since it disregards time axis misalignments. Figure 5a illustrates this problem. While Euclidean distance considers the same time indexes for computing the dissimilarity (subscripts i in Equation 9), if there is misalignment in time, the distance is greater than it would be in case the offset was zero. Figure 5b presents the alignment considered by Dynamic Time Warping, which addresses this drawback.

$$dist_{x,y} = \left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{1/p} \tag{9}$$

5.1.2 Dynamic Time Warping The Dynamic Time Warping (DTW) distance [5], considers time axis offsets when comparing two time series, as presented in Figure 5. Because



(a) Alignment according to Euclidean distance.

(b) Alignment according to Dynamic Time Warping.

Figure 5: Sample case of time axis misalignment, which is not considered by the Euclidean distance. Other dissimilarities, such as Dynamic Time Warping, are capable of detecting this problem and not penalize relations between the series.

it is a more robust measure than Euclidean distance and having been widely employed in the literature, it was also selected for evaluation.

The computation of Dynamic Time Warping distance works as follows [23]. Consider two time series $X = x_1, x_2, \ldots, x_n$ and $Y = y_1, y_2, \ldots, y_m$. To align both sequences, a matrix $D_{n \times m}$ is constructed, where element $d_{i,j}$ contains the distance $d(x_i, y_j) = (x_i - y_j)^2$ between points. Each element (i, j) is an alignment of points x_i and y_j. A warping path W is a traversing of contiguous elements in the matrix, which defines a mapping between X and Y. The k-th element of W is defined as $w_k = (i, j)_k$. Having that, $W = w_1, w_2, \ldots, w_K$ is obtained, with $max(m, n) \leq K < m + n - 1$. Generally, the path in the matrix is a diagonal, starting in opposite extremes of the matrix, for example $w_1 = (1, 1)$ and $w_K = (m, n)$. Element w_k has to be adjacent to w_{k-1}, i.e., it is only allowed to select elements in the path which are adjacent. The points in W must also be monotonically spaced in time. The objective is to select the path that minimizes the cost of warping, according to Equation 10.

$$DTW(X, Y) = \min\left(\sqrt{\sum_{k=1}^{K} w_k}\right) \tag{10}$$

The evaluation of warping paths by DTW makes it possible to find the best synchronization of both series, which allows the dissimilarity to be more robust to time axis misalignments. Other commonly used dissimilarity measures are based on correlations, discussed next.

5.1.3 1 - Pearson Correlation Distance The correlation proposed by Pearson [24] computes the linear relationship between sequences of numbers, taking into account their magnitudes. Equation 11 presents Pearson's r coefficient, in which $\overline{x}$ and $\overline{y}$ are the averages of sequences X and Y respectively. The values returned by Pearson's r stay in interval $[-1, 1]$, simplifying interpretation and comparison. When the series are independent, their correlation tends to zero. Conversely, when there is a perfect positive or negative linear relationship, it tends to $|r| = 1$.

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{11}$$

Pearson correlation provides a measure of similarity between series. Because we wish to compare time series based on their distances (or dissimilarities), the transformation $d_{Pearson} = 1 - r$ is used.

A problem with Pearson correlation is its sensitivity to noise and outliers [25], besides being only capable of detecting linear relationships. Another measure, which minimizes these problems, is Spearman correlation.

5.1.4 1 - Spearman Correlation Distance Spearman [26] is a specialization of Pearson's correlation, which considers only sequence ranks. That makes the measure more robust to the presence of noise and outliers, as the magnitudes of the numbers are disregarded. When there is a monotonically increasing or decreasing relation, Spearman is also able to detect nonlinear similarities.

The value of Spearman correlation is computed in the same way as Pearson (Equation 11), however only the ranks of series elements are used. When there are ties among two or more values, the ranks assigned to the tied elements are the sum of the ranks divided by the number of repeated elements. For example, the sequence $X = (1, 3, 3, 10)$ has $rank(X) = (1, 2.5, 2.5, 4)$.

As Spearman correlation is a measure of similarity, we use the same transformation used for Pearson, which returns a dissimilarity (Equation 12).

$$d_{Spearman} = 1 - r(rank(X), rank(Y)) \tag{12}$$

5.2 1 - Cosine similarity

The cosine similarity computes the cosine of the angle between two vectors in space [7]. This can be interpreted as a way to check if both vectors point to the same direction. The similarity is defined in the interval [-1,1], thus for turning it into a dissimilarity measurement we convert it by rescaling into the range [0,1] and then subtracting from 1, as in Equation 13 for two vectors X and Y.

$$d_{cos} = 1 - \frac{\frac{X \cdot Y}{||X|| \cdot ||Y||} + 1}{2} \tag{13}$$

The cosine similarity is useful for assymetric, non-binary attributes. Its most common application is in text mining with word frequency tables, in which the absence of words in a document should not contribute to the similarity calculation, but only the words that do appear in the document. Although not a common similarity measure for time series, we also test it because of its popularity in machine learning.

5.2.1 Dist_pacf distance The dissimilarity measures presented so far are commonly employed in the literature. However, they do not consider the inherent correlations of time series, but analyze only one compared to the others.

In this paper, we propose the use of a measure that takes into account serial correlations. For that, the partial autocorrelation function, explained in Section 3, was used to define a new dissimilarity measure.

The measure is described by Equation 14, in which $pacf$ is the partial autocorrelation of the series, δ is a parameter used by function $first$, which returns the initial δ partial autocorrelations of the series (excluding the one with $\tau = 0$, as it is always one). After obtaining these sequences of δ partial autocorrelations, the Manhattan distance [7] between them is computed, which is simply the Minkowski norm with $p = 1$ (Equation 9).

$$\begin{aligned} Dist\_pacf_{\delta}(X,Y) = Manhattan(&first(pacf(X),\delta), \\ &first(pacf(Y),\delta)) \end{aligned} \tag{14}$$

The intuition behind this dissimilarity measure is that it initially evaluates the partial autocorrelations of each series, with $\tau = 1, 2, \ldots, \delta$, obtaining a better characterization of which regressive model it follows. Next, with these new sequences, $first(pacf(X), \delta)$ and $first(pacf(Y), \delta)$, obtained for X and Y, their distance can be computed. The value returned provides an estimate of how close are the regressive models that generated both series.

In the next section, experimental results with the hierarchical clustering algorithms and dissimilarities discussed are presented.

6 Experimental Results

Experiments were executed in a machine with an Intel Core(TM) i7 @2.80Hz, 4 GB of RAM and Ubuntu 11.04 operating system. The implementations were done in the statistical computing software R 2.12.1 [27].

6.1 Experiments with synthetic data

In this section, results concern the synthetic bases generated with series following Box-Jenkins AR models and the Gaussian data bases.

Table 3 presents the results with the execution of the algorithms and respective dissimilarities on the $AR(p)$ data bases. The results are generally around zero, according to the Adjusted Rand Index. That indicates low quality clustering, comparable to what would be obtained with a random data partitioning. In order to better evaluate these results, a new set of experiments was executed using the Gaussian data bases, which contained statistically independent data.

Table 3: Results (ARI) for synthetic $AR(p)$ data bases.

Dissimilarity	**Single-linkage**		**Complete-linkage**		**Average-linkage**	
	Mean	Stdev	Mean	Stdev	Mean	Stdev
Euclidean	0.015	0.017	0.097	0.085	0.036	0.054
DTW	0.016	0.026	0.071	0.101	0.040	0.049
$1-$ Pearson	0.010	0.048	-0.009	0.080	-0.006	0.064
$1-$ Spearman	0.017	0.072	0.001	0.091	0.003	0.070
$1-$ Cosine	0.021	0.061	0.012	0.089	0.018	0.091
$Dist\_pacf_\delta$	0.328	0.171	0.416	0.154	0.402	0.146

Table 4: Results (ARI) for synthetic $N(\mu, \sigma)$ data bases.

Dissimilarity	**Single-linkage**		**Complete-linkage**		**Average-linkage**	
	Mean	Stdev	Mean	Stdev	Mean	Stdev
Euclidean	0.177	0.222	0.683	0.211	0.630	0.220
DTW	0.191	0.228	0.711	0.214	0.620	0.220
$1-$ Pearson	-0.007	0.038	-0.009	0.073	-0.017	0.083
$1-$ Spearman	-0.005	0.025	0.016	0.085	0.001	0.084
$1-$ Cosine	0.021	0.006	0.065	0.048	0.023	0.000

Table 4 presents the results using the Gaussian data bases. Results were substantially better with Complete-linkage and DTW dissimilarity. This shows that traditional clustering algorithms, along with the commonly used dissimilarity measures, can find good partitions when observations are independent and identically distributed, which for time series is not a valid assumption, as they tend to be correlated.

To show how these results can be improved, taking into account the temporal relationship of the data, we proposed a dissimilarity measure, Dist_pacf, based on series partial autocorrelations, described in Section 5.2.1.

Parameters $\tau = 20$ and $\delta = 3$ were used, as the synthetic series belonged to at most an $AR(3)$ process. Results of the proposed dissimilarity are presented in Table 3. The ARI values are substantially greater than the other dissimilarities on the $AR(p)$ bases.

We can also observe the improvement in the results by analyzing the dendrograms. Figure 6a presents the dendrogram for *Complete-linkage* with Dynamic Time Warping, while Figure 6b presents the dendrogram for *Complete-linkage* with Dist_pacf. Several objects pertaining to the same cluster were joined together first in the hierarchy using the proposed dissimilarity measure.

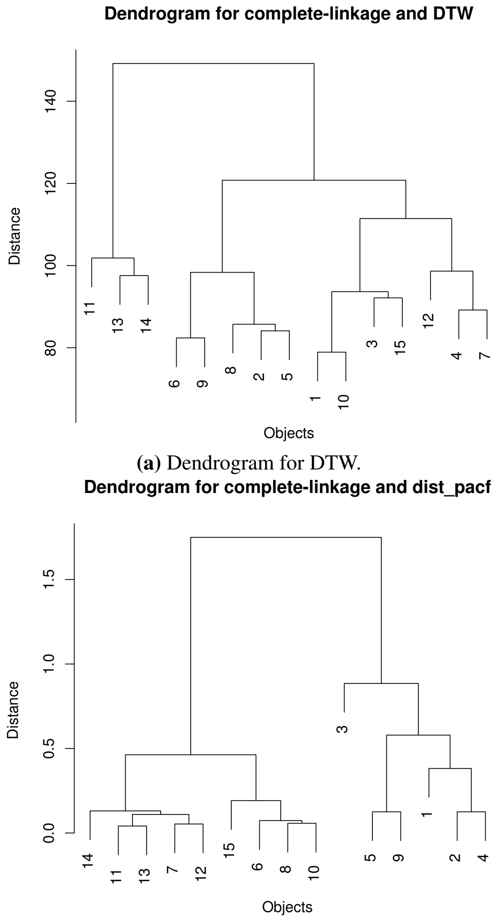
Another way to visualize the improvement of considering the serial correlations present in data is by looking at the dissimilarity matrix generated by Dist_pacf. Figure 7a presents the matrix for DTW, while Figure 7b presents the dissimilarity matrix for Dist_pacf. Ideally, it is expected that clusters form well defined blocks, as illustrated in Figure 7c. We observe there is no cluster structure in the DTW matrix. On the other hand, Dist_pacf provides a better identification of a clustering structure, although not perfect. It should be observed, however, that the models used for the clusters were close to each other, in terms of regressive coefficients (cf. Table 2), making it harder to obtain a high-quality crisp clustering.

6.2 Experiments with real data

Experiments using real data were also performed, employing physiological sensor data from several individuals, as previously discussed in Section 5. Since in this scenario there was no previously known ideal partition, the Adjusted Rand Index was not used for validation. Instead, the Normalized Hubert Γ statistic was selected, which was discussed in Section 5.

The results for these experiments are presented in Table 5. We observe greater values of Γ are obtained, however, we note that the Γ statistic measures the correlation between the cophenetic matrix and the proximity matrix, which tends to be high for a reasonable clustering algorithm.

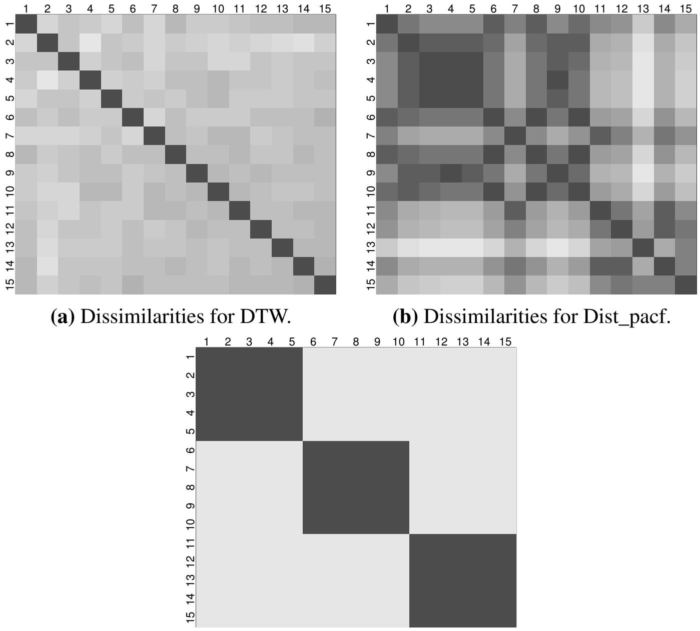
Figure 8a presents the dendrogram for the first data base using Euclidean distance. We



(a) Dendrogram for DTW.

(b) Dendrogram for Dist_pacf.

Figure 6: The dendrograms show a comparison of *Complete-linkage* using Dynamic Time Warping and Dist_pacf. The clusters in this AR base were given by $\{\{x_1, x_2, x_3, x_4, x_5\}, \{x_6, x_7, x_8, x_9, x_{10}\}, \{x_{11}, x_{12}, x_{13}, x_{14}, x_{15}\}\}$. Using DTW, the objects are merged disregarding their correct cluster. With Dist_pacf, several objects were firstly merged with other objects from their own cluster.



(a) Dissimilarities for DTW.

(b) Dissimilarities for Dist_pacf.

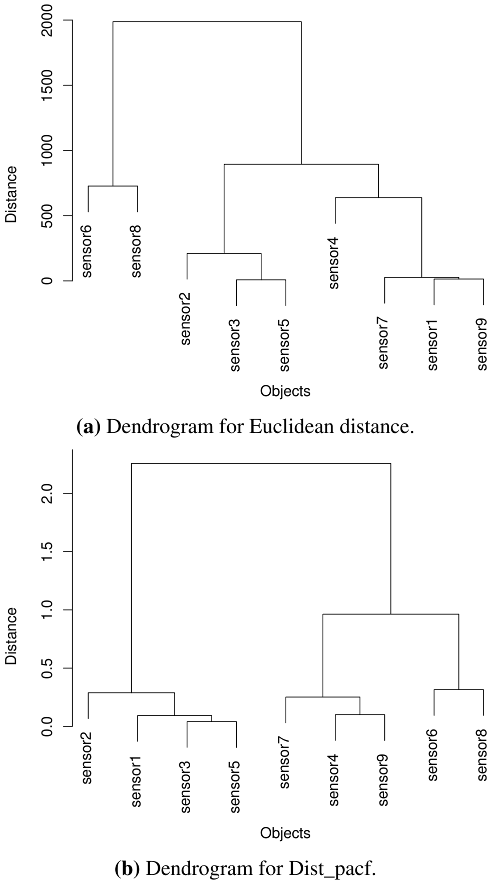
(c) Ideal dissimilarities.

Figure 7: The matrices represent the level of dissimilarity measured between elements (i, j) according to some distance. Darker gray indicates less distance (or more similarity). The visualization allows checking whether the dissimilarity measure is adequately representing the clustering structure present in data. For DTW (a), we observe there is virtually no cluster structure being identified. For Dist_pacf (b), the original cluster structure was partially retrieved. In (c), we see the ideal cluster structure.

Table 5: Results (Γ) for real ICML 2004 data bases.

Dissimilarity	**Single-linkage**		**Complete-linkage**		**Average-linkage**	
	Mean	Stdev	Mean	Stdev	Mean	Stdev
Euclidean	0.959	0.028	0.961	0.025	0.966	0.020
DTW	0.855	0.085	0.876	0.062	0.892	0.045
$1-$ Pearson	0.952	0.030	0.948	0.031	0.963	0.022
$1-$ Spearman	0.947	0.031	0.945	0.031	0.960	0.021
$1-$ Cosine	0.982	0.051	0.983	0.045	0.990	0.017
$Dist\_pacf_\delta$	0.903	0.060	0.913	0.050	0.922	0.042

note that some similar sensors were merged together first, as expected. For example, referring back to Table 1 in Section 5, we see that sensors 2, 3 and 5 are related to the individual's temperature. Sensors 6 and 7 are related to the longitudinal accelerometer, while 8 and 9 to the transverse. It is expected that those sensors are merged together first in the hierarchy produced by the clustering algorithms, as they provide related information. Figure 8b presents the dendrogram for Dist_pacf. We observe that several sensors with related characteristics were correctly merged together in a first stage.



(a) Dendrogram for Euclidean distance.

(b) Dendrogram for Dist_pacf.

Figure 8: Dendrograms obtained for the first ICML 2004 data base. We observe several sensors with similar functionality (see Table 1 in Section 5) were firstly merged using both dissimilarity measures.

7 Discussion

The experiments presented in this paper show that traditional dissimilarity measures do not produce satisfactory results for time series clustering, as they disregard autocorrelations. To overcome this drawback, this paper proposes a new dissimilarity measure, Dist_pacf, based on partial autocorrelations.

Experiments using synthetic data sets with time series simulated from Autoregressive processes showed that Dist_pacf has better results than other commonly used dissimilarity measures (see Table 3). This can be explained by the fact that most of the commonly used dissimilarities completely disregard the intrinsic correlations among observations of a time series. Most machine learning methods make the assumption that data are independent and identically distributed, which is not the case with time series. This fact was illustrated in Figure 4, which shows the strength of correlations for an AR and a Gaussian (i.i.d.) data base, indicating that for proper time series data there is strong correlation present.

This limitation found on the common dissimilarity measures used for time series, such as Dynamic Time Warping (DTW), can be easily observed in Figure 7, which presents a gray-colored matrix with darker tones indicating more similarity. Ideally, for data with a clear cluster structure and three clusters, there should be three well defined blocks of darkened cells. In Figure 7 we see that DTW is unable to find any cluster structure in the AR data base. The only darkened coloured cells are the ones in the main diagonal, which relates examples to themselves. Our proposed dissimilarity measure was able to partially recover the cluster structure. A perfect reconstruction was not possible because the autoregressive models of the time series were close to each other, differing only by one or two autoregressive coefficients.

The experiments with real data sets from the ICML 2004 challenge showed that Dist_pacf had competitive results as the other dissimilarity measures. It is important to note why the other measures had better results than for the synthetic data bases. This can be explained because of the evaluation measure used. Since for this data set we do not have an external correct partition as reference, the Adjusted Rand Index could not be used, so the Normalized Hubert Γ statistic was chosen as the evaluation metric, which measures the correlation between the cophenetic and proximity matrices. Because it is a simple correlation of both matrices and the cophenetic matrix is constructed using the proximity matrix as input, reasonable clustering algorithms will return good values for the statistic, even though the original clusters may not be fully reconstructed.

Overall, our proposed dissimilarity measure is a step forward in modelling time series autocorrelations, which are a useful indicator of the generating model, or equation, which produced the time series. This is achieved by direct use of the Partial Autocorrelation Function (PACF), which computes how much and how far time series observations are related to one another, as discussed in Section 5.2.1

The limitations of the proposed measure are: 1) the determination of parameter τ, the maximum lag to compute the partial autocorrelations; 2) the choice of parameter δ, for the subselection of the first partial autocorrelations; 3) partial autocorrelations depend on Pearson's r coefficient, which is limited to representing linear relationships.

We note that, in practice, if the desired regressive extension δ is known, then one can simply set $\tau = \delta$ and avoiding the computational effort to determine autocorrelations at farther lags.

Other modeling possibilities do exist, such as fitting AR models, using a selection criteria, e.g., Akaike Information Criteria (AIC) [28]. That, however, restricts the application of the method to certain fitting models. A series that follows an ARIMA model, for example, would not be adequately represented if the fitting models do not consider non-stationary relations.

8 Conclusion

Our findings show that the choice of an adequate dissimilarity measure is crucial for time series clustering. The popular distance functions, such as Euclidean, DTW, and Pearson-based ones, completely disregard inherent autocorrelations present in series. While the choice of a clustering algorithm is also an important decision, we claim that the selection of a dissimilarity measure plays an even bigger role, as confirmed by our experiments. If there is no clustering structure provided by the proximity matrix, then even the best clustering algorithm is unable to find a correct data partition. In summary, we proposed a new dissimilarity measure which takes into account series partial autocorrelations to overcome the main drawback found in the most popular ones. This research indicates that future endeavors in non-parametric dissimilarity measures, capable of also dealing with nonlinear autocorrelations, are promising research directions.

References

[1] Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988. ISBN 0-13-022278-X.

[2] B.S. Everitt, S. Landau, M. Leese, et al. *Cluster analysis*. Edward Arnold, London, 2001.

[3] R. Xu and D. Wunsch. *Clustering*. Wiley-IEEE Press, 2008.

[4] Eamonn Keogh and Shruti Kasetty. On the need for time series data mining benchmarks: a survey and empirical demonstration. In *Proceedings of the eighth ACM SIGKDD*

international conference on Knowledge discovery and data mining, KDD '02, pages 102–111, New York, NY, USA, 2002. ACM. ISBN 1-58113-567-X.

[5] D. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop on Knowledge Discovery in Databases*, volume 10, pages 359–370, 1994.

[6] G.E.P. Box and G.M. Jenkins. *Time series analysis: forecasting and control*. Prentice Hall PTR, 1994.

[7] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005. ISBN 0321321367.

[8] Pedro A. Morettin and Clélia M. C. Toloi. *Análise de Séries Temporais*. Associação Brasileira de Estatística (ABE), 2a edição edition, 2006. ISBN 978-85-212-0389-6.

[9] Rakesh Agrawal, Christos Faloutsos, and Arun N. Swami. Efficient similarity search in sequence databases. In *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*, FODO '93, pages 69–84, London, UK, UK, 1993. Springer-Verlag. ISBN 3-540-57301-1.

[10] Rakesh Agrawal, King-Ip Lin, Harpreet S. Sawhney, and Kyuseok Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Proceedings of the 21th International Conference on Very Large Data Bases*, VLDB '95, pages 490–501, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1-55860-379-4.

[11] Béla Bollobás, Gautam Das, Dimitrios Gunopulos, and Heikki Mannila. Time-series similarity problems and well-separated geometric sets. In *Proceedings of the thirteenth annual symposium on Computational geometry*, SCG '97, pages 454–456, New York, NY, USA, 1997. ACM. ISBN 0-89791-878-9. doi: 10.1145/262839.263080.

[12] Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proc. VLDB Endow.*, 1(2):1542–1552, August 2008. ISSN 2150-8097.

[13] Byoung-Kee Yi and Christos Faloutsos. Fast time sequence indexing for arbitrary lp norms. In *Proceedings of the 26th International Conference on Very Large Data Bases*, VLDB '00, pages 385–394, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1-55860-715-3.

[14] E. Frentzos, K. Gratsias, and Y. Theodoridis. Index-based most similar trajectory search. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 816 –825, april 2007.

[15] Henrik André-Jönsson and Dushan Z. Badal. Using signature files for querying time-series data. In *Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery*, PKDD '97, pages 211–220, London, UK, UK, 1997. Springer-Verlag. ISBN 3-540-63223-9.

[16] Lei Chen and Raymond Ng. On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, VLDB '04, pages 792–803. VLDB Endowment, 2004. ISBN 0-12-088469-0.

[17] Lei Chen, M. Tamer Özsu, and Vincent Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, SIGMOD '05, pages 491–502, New York, NY, USA, 2005. ACM. ISBN 1-59593-060-4. doi: 10.1145/1066157.1066213.

[18] Michael D. Morse and Jignesh M. Patel. An efficient and accurate method for evaluating time series similarity. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, SIGMOD '07, pages 569–580, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-686-8.

[19] Johannes Aßfalg, Hans-Peter Kriegel, Peer Kröger, Peter Kunath, Alexey Pryakhin, and Matthias Renz. Similarity search on time series based on threshold queries. In *Proceedings of the 10th international conference on Advances in Database Technology*, EDBT'06, pages 276–294, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-32960-9, 978-3-540-32960-2.

[20] Yueguo Chen, M.A. Nascimento, Beng Chin Ooi, and A.K.H. Tung. Spade: On shape-based pattern detection in streaming time series. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 786–795, april 2007. doi: 10.1109/ICDE.2007.367924.

[21] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 262–270, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1462-6. doi: 10.1145/2339530.2339576.

[22] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985. ISSN 0176-4268. 10.1007/BF01908075.

[23] Eamonn Keogh and Chotirat Ann Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7:358–386, 2005. ISSN 0219-1377. 10.1007/s10115-004-0154-9.

[24] K. Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110, 1894.

[25] A. Fujita, J.R. Sato, MA Demasi, M.C. Sogayar, C.E. Ferreira, and S. Miyano. Comparing pearson, spearman and hoeffding's d measure for gene expression association analysis. *Journal of Bioinformatics and Computational Biology*, 7(4):663–84, 2009.

[26] C. Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 15(1):72–101, 1904.

[27] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

[28] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.