Time Series Classification Using Compression Distance of Recurrence Plots

Diego F. Silva, Vinícius M. A. de Souza, Gustavo E. A. P. A. Batista
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
São Carlos, Brazil
{diegofsilva,vsouza,gbatista}@icmc.usp.br

***Abstract*—There is a huge increase of interest for time series methods and techniques. Virtually every piece of information collected from human, natural, and biological processes is susceptible to changes over time, and the study of how these changes occur is a central issue in fully understanding such processes. Among all time series mining tasks, classification is likely to be the most prominent one. In time series classification there is a significant body of empirical research that indicates that k-nearest neighbor rule in the time domain is very effective. However, certain time series features are not easily identified in this domain and a change in representation may reveal some significant and unknown features. In this work, we propose the use of recurrence plots as representation domain for time series classification. Our approach measures the similarity between recurrence plots using Campana-Keogh (CK-1) distance, a Kolmogorov complexity-based distance that uses video compression algorithms to estimate image similarity. We show that recurrence plots allied to CK-1 distance lead to significant improvements in accuracy rates compared to Euclidean distance and Dynamic Time Warping in several data sets. Although recurrence plots cannot provide the best accuracy rates for all data sets, we demonstrate that we can predict ahead of time that our method will outperform the time representation with Euclidean and Dynamic Time Warping distances.**

I. Introduction

In the last years, the Data Mining community has witnessed a huge increase of interest for time series methods and algorithms. Such interest is justified by the innumerous applications that generate data across time. Virtually every piece of information collected from human, natural, and biological processes is susceptible to changes over time. The study of how these changes occur is a central issue in fully understanding such processes.

Among all time series mining tasks, classification is likely to be the most prominent one. In this task, we are interested in associating a discrete class to individual time series. A simple and effective procedure for time series classification is similarity search. For instance, the k-nearest neighbor rule (k-NN) uses a distance function $d(t, q)$ between two time series t and q, to find the k most similar training instances $t_1, t_2, \dots t_k$ to a query instance q. The class mode among the k most similar instances is then predicted to q.

There is a significant body of empirical research that indicates that similarity search is very effective to time series classification (see, for instance, [12], [36]). These studies usually use a distance function in the time domain to measure the similarity between time series. Two distance measures commonly used in time series classification are Euclidean distance (ED) and Dynamic Time Warping (DTW). DTW can be understood as an extension of Euclidean distance able to provide nonlinear time scaling invariance, popularly known as warping [5].

However, certain time series features are not evident in the time domain. One example is sound recognition in which the features are usually identified in the frequency domain. There is a large number of signal processing methods that promote a change of representation and identify features in power spectrum, cepstrum or spectrogram [3], [33], [16]. Although frequency is likely to be the most explored alternative representation domain, other possibilities also evaluated in the literature are wavelets, principal component analysis, autocorrelation [1], shapelets [26], etc.

In this work we propose the use of recurrence plots as representation domain for time series classification. Recurrence plots are widely used techniques for qualitative assessment of time series in dynamical systems. Their graphical nature exposes hidden patterns and structural changes in data. In particular, recurrence plots are outstanding tools to characterize how the similarity among subsequences varies according to time.

We evaluate the hypothesis that such information can generally help the classification of time series in a wide range of application domains. The intuition behind our proposal is that recurrent patterns are regularities frequently associated with interesting behaviors. A recurrent behavior indicates the presence of an internal mechanism that generates such patterns, opposed to a random (and uninteresting) series in which no patterns are present. The explicit representation of such regularities can reveal the underlining mechanisms that generated the data, and thus is a potentially useful feature to classify time series.

Our approach uses the Campana-Keogh (CK-1) distance [8] to measure the similarity between recurrence plots. CK-1 is a Kolmogorov complexity-based distance that uses video compression algorithms to estimate image similarity. We show that the recurrence plots allied to CK-1 lead to significant improvements in accuracy rates compared to ED and DTW in several data sets from the UCR archive. Although recurrence plots cannot provide the best accuracies for all data sets, and the central assumption of this work is that no single representation is best for every domain, we demonstrate that we can predict ahead of time that our method will outperform the time representation with the aforementioned distance measures.

In order to achieve our goals, we start describing the basic concepts present in the proposed method, in Section II, followed by the description of relevant related work in Section III. In Section IV we describe the proposed algorithm, and present our experimental setup and results in Section V. Finally, we conclude our work and present directions for future research in Section VI.

II. Background

This section reviews recurrence plots and the CK-1 distance measure. We will also briefly discuss the Euclidean and DTW distances since they are used in our experimental evaluation.

A. Recurrence Plots

The relevance of recurrent behaviors, such as seasonality, in natural processes has been studied for decades [28]. However, the visualization of these behaviors often are very difficult in the time domain. To overcome this limitation, Eckmann et al. [14] created a representation called Recurrence Plot (RP). This tool allows for the investigation of m-dimensional trajectories in a bi-dimensional phase space. This representation is able to reveal in which points these trajectories return to a previously visited state.

Formally, a RP can be defined according to Equation 1.

$$R_{i,j} = \Theta(\epsilon - ||\vec{x}(i) - \vec{x}(j)||), \vec{x}(\cdot) \in \Re^m, i,j = 1..N \quad (1)$$

where N is the number of states, $\vec{x}_i$ and $\vec{x}_j$ are the subsequences observed at the positions i and j, respectively, $||\cdot||$ is the norm (e.g. Euclidean norm) between the observations, ϵ is a threshold for closeness and Θ is the Heaviside function, defined by Equation 2.

$$\Theta(z) = \begin{cases} 0, & \text{if } z < 0 \\ 1, & \text{otherwise} \end{cases} \quad (2)$$

Equation 1 states that if the m-dimensional trajectory of the time series at time j is close (in terms of a pre-defined neighborhood) to the subsequence observed at time i, there will be a value 1 at position (i,j) of recurrence matrix. Otherwise, the value is 0. In the graphical representation, an image of $N \times N$ pixels is defined so that the pixels corresponding to values 1 in the matrix are commonly black and the 0's are white. Figure 1 shows a few examples of recurrence plots for signals with different degrees of randomness.

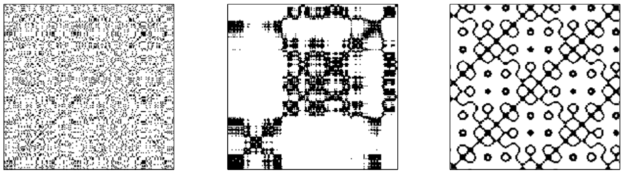

Fig. 1. Some examples of recurrence plots: totally random noise (*left*); random walk (*middle*); periodic composition of sine and cosine (*right*)

Despite its simplicity, this method requires the specification of a closeness threshold parameter, which defines the size of a neighborhood in which two subsequences are considered similar. However, determining an appropriate value for this parameter is not intuitive. The practice has come out with a few heuristics. For instance, a threshold of 10% of the largest observed distance, or a value that results in a certain percentage of black points. However, these are local heuristics, i.e., they use information of a single recurrence plot to set the threshold value. Therefore, it is difficult to generalize a threshold value that is consistent according multiple recurrence plots. This is an important issue when we want to determine the similarity between two recurrence plots.

In order to eliminate the closeness parameter, we can make use of color information. The image is generated with grayscale or other color maps so that the distances are represented as color. Thus, the image is a direct representation of the distance matrix. The recurrence plot is no longer a tool to analyze recurrences considering neighborhoods. It becomes a tool to analyze how close each pair of subsequences are in their trajectories [18]. This representation is known as unthresholded recurrence plot, distance plot or self-similarity matrix. Figure 2 shows an example of a thresholded and an unthresolded recurrence plot for a same time series.

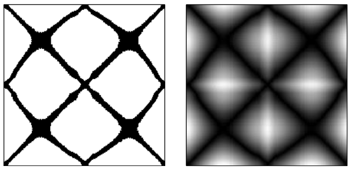

Fig. 2. Examples of a thresolded (*left*) and an unthresholded (*right*) recurrence plots, both generated from a same time series

The definition of Recurrence Plots in Equation 1 has a second parameter m that defines the dimension of the trajectory. In our experiments, we chose $m = 1$ for all data sets. This means that, although we may consider m consecutive points to analyze the trajectory, our experiments just use one-dimensional trajectory. We chose $m = 1$ since the structure of the recurrence plots seems to change very little as we vary such parameter. Our opinion is also suported by the recurrence plot literature. In the most comprehensible study regarding this parameter of our knowledge, Iwanski and Bradley affirm that "*while examining several recurrence plots of a particular data set, we noticed that their appearances seemed to remain qualitatively unchanged with changing embedding dimension*" [18].

We note that this one-size-fits-all aproach may not provide the best classification results for all data sets, and tuning this parameter on training data may improve our classification results. Therefore, the experimental comparison with Euclidean distance and Dynamic Time Warping of Section V might be slightly pessimistic for our method.

B. Normalized Compression and Campana-Keogh Distances

Campana-Keogh (CK-1) is a recently proposed distance to estimate the similarity between two images [8]. The main theoretical basis for the CK-1 measure is the concept of the

Kolmogorov complexity. The Kolmogorov complexity $K(x)$ of a string x is defined as the length of the shortest program capable of producing x on a universal computer, such as a Turing machine [24]. Intuitively, $K(x)$ is the minimal quantity of information required to generate a string x with a program.

The notion of conditional complexity is necessary to define a distance based on the Kolmogorov complexity. In [23], the authors define a distance between two strings x and y according to Equation 3.

$$d_k(x, y) = \frac{K(x|y) + K(y|x)}{K(xy)} \tag{3}$$

where $K(x|y)$ is defined as the length of shortest program that outputs x, given y as auxiliary input.

Although Kolmogorov conditional complexity gives rise to a distance measure that is optimal in the sense that it subsumes other measures, such a distance is uncomputable in the general case. Therefore, several researchers have proposed approximations to this distance using compression algorithms [25], [19] and many others have evaluated these approximations in diverse domains [22], [7].

Given a data compression algorithm, we define $C(x)$ as the size of the compressed x and $C(x|y)$ as the compression size of x after training the compressor in y [8]. We can calculate a compression approximation of the Kolmogorov complexity distance defined in Equation 3 using Equation 4.

$$d_c(x, y) = \frac{C(x|y) + C(y|x)}{C(xy)} \tag{4}$$

The better the compression algorithm, the better the approximation of d_c for d_k is [19].

One of the best-known distances that make use of a compression approximation of Kolmogorov complexity is the normalized compression distance (NCD) [25]. It is defined according Equation 5.

$$d_{ncd}(x, y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}} \tag{5}$$

NCD has been successfully applied for measuring the similarity between two sequences in a number of application domains. For instance, [25] uses NCD to construct the phylogeny tree based on whole mitochondrial genomes, and a language tree for over 50 Euro-Asian languages.

Although NCD is well-suited for comparing sequences, its use for comparing images would require image linearization; incurring in some spatial information loss. CK-1 extends the applicability of compression-based distances to images by using video compression. Given two images x and y, CK-1 is defined by Equation 6.

$$d_{mpeg}(x, y) = \frac{C(x|y) + C(y|x)}{C(x|x) + C(y|y)} - 1 \tag{6}$$

where $C(a|b)$ is the size of a synthetic MPEG-1 video composed by two frames $b \in \{x, y\}$ and $a \in \{x, y\}$, in this order.

MPEG-1, and most video encoding algorithms, achieve compression by finding recurring patterns within a frame (intra frame compression) and/or between frames (inter frame compression). When x and y are two similar images, inter frame compression step should be able to exploit that to produce a smaller file size, which can be interpreted as significant similarity. As digital video is an important commercial application, many efforts have been made to achieve high compression rates in video encoding, making it a good approximation of the Kolmogorov conditional complexity. Another observation is that there is no necessity to "*hack*" the video encoding algorithm, since no internal modification is necessary.

C. Time Series Classification with ED and DTW

The k-nearest neighbors algorithm works with the intuitive idea that if two series are similar, they are likely to have the same class. In order to classify a query time series, the algorithm measures the dissimilarity between the query and all labeled series in the training set. The class assigned to the query time series is the class mode among the k most similar training instances.

The k-NN algorithm leaves open the choice of a distance measure. A simple and widely used measure is the Euclidean distance, defined in Equation 7.

$$ED(x, y) = \sqrt{\sum_{t=1}^{n} (x(t) - y(t))^2} \tag{7}$$

The Euclidean distance measures the similarity considering two observations in the exact same moment t. However, many applications require a more flexible matching of observations, in which an observation of x at time t may be matched to an observation of y in a time near to t. The Dynamic Time Warping (DTW) distance achieves an optimal nonlinear alignment of the observations under some constraints. Figure 3 illustrates the difference between the linear alignment obtained by the Euclidean distance, and the nonlinear alignment obtained by the DTW algorithm.

DTW provides the smallest distance obtained by allowing a nonlinear matching of the observations under the following constraints:

- **Boundary constraint.** The matching is made for the whole time series x and y, therefore it starts at $(1, 1)$ and ends at (n, m);
- **Continuity constraint.** The matchings are made in one-unit steps. It means that the matching never jumps one or more observations;
- **Monotonicity constraint.** The relative order of observations has to be preserved.

DTW is usually calculated using a dynamic programming algorithm. The algorithm is based on the initial condition described in Equation 8.

$$DTW(i, j) = \begin{cases} 0, \text{ if } i, j = 0 \\ \infty, \text{ otherwise} \end{cases} \tag{8}$$

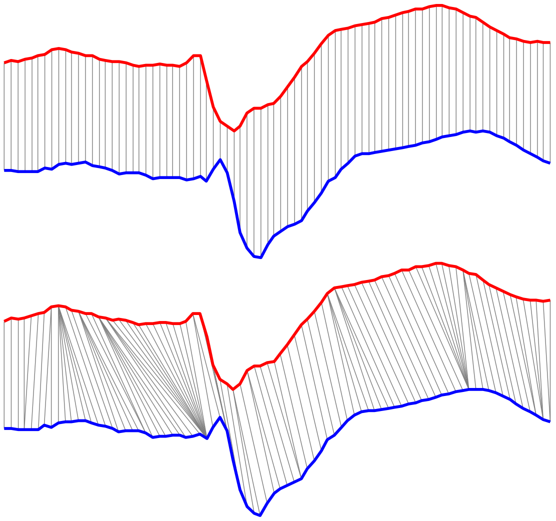

Fig. 3. Difference between the alignment used by Euclidean distance (*top*) and DTW (*bottom*)

The recurrence relation of DTW algorithm is presented in Equation 9.

$$DTW(i,j) = C(x(i), y(j)) + min \begin{cases} DTW(i-1,j) \\ DTW(i,j-1) \\ DTW(i-1,j-1) \end{cases} \tag{9}$$

where $i = 1 \dots n$ and $j = 1 \dots m$ and n and m are the lengths of the x and y time series, respectively. $C(x(i), y(j))$ is the cost of matching two observations $x(i)$ and $y(j)$, frequently calculated with Euclidean distance. The resulting value in $DTW(n,m)$ is the DTW distance between x and y.

III. Related Work

Recurrence plots are graphical representations of time series. They were introduced by Eckmann et al. [14], and are frequently used in time series research and applications. A few instances are the analysis of social insects behavior [29], protein structural prediction [31], stock market analysis [2], and multimodal communicative signals analysis [10]. One field of study that the use of recurrence plot is relatively common is the analysis of biological signals. Some examples are the analysis of electrocardiograms [34], electroencephalograms [9] and electromyograms [15], [30], as well detection of coronary artery disease [13].

In general, classification using recurrence plots is made by extracting local features from the plots that try to quantify the small scale structures. Several researchers have proposed measures based on recurrence point density and diagonal and vertical line structures. Many of these measures are detailed surveyed in [27]. In this work, we use an orthogonal approach and we do *not* search for local features. Instead, we compare whole recurrence plots using a compression distance. We believe our approach is simpler and has the additional advantages of being feature set and domain independent and parameter-free.

The research work that has most similarity to ours is [6]. The author uses thresholded recurrence plots of chroma features (musical chord sequences representations) as a structural representation in music. Normalized Compression Distance (NCD) [25] was used to measure the distance between the plots. However, as we noticed before, NCD uses standard compression algorithms that are well-suited for compression of discrete sequences, such as strings. These algorithms look for exact recurrent subsequences and replace those subsequences in order to gain compression. However, exact recurrent subsequences are rare in real-valued data, such as the color information of images, making those algorithms an unsuitable approximation to Kolmogorov complexity of images. In addition, standard compression algorithms work with sequences and are not able to make use of the spatial information of images.

In contrast, CK-1 is more suitable distance between the real-valued matrices that represent the recurrence plot images. This distance measure has proven to be very effective in several tasks. For instance, it was used in to successfully classify moths [4], to identify sounds generated by insects using spectrograms [16] and to analyze digitalized images of stylized letters used in the beginning of each chapter in historical texts [17]. In Section IV we detail our proposed method.

IV. Proposed Method

The intuition behind our proposal is that recurrent patterns are regularities frequently associated to interesting behaviors. A recurrent behavior indicates the presence of an internal mechanism that generates such patterns, opposed to a completely random (and uninteresting) series in which no patterns are present. The explicit representation of such regularities can reveal the underlining mechanisms that generated the data, and thus it is a potentially useful feature to classify time series.

We evaluate the hypothesis that such information can generally help the classification of time series in a wide range of application domains. We raised this hypothesis by observing that time series from different classes frequently present recurrence plots that can be easily separated by eye. In contrast, time series from a same class frequently present recurrence plots with strong apparent similarity. This observation can be illustrated with a simple experiment shown in Figure 4. In this experiment, we chose a random sample of six leaves from the *Swedish Leaf* data set that were incorrectly classified by 1-nearest neighbor classifier with Euclidean distance but correctly classified by measuring the distance of their recurrence plots.

Notice in Figure 4 that although all time series have a similar "w" shape, the recurrence plots for objects from different classes can be easily distinguished by eye. In addition, the recurrence plots for objects of the same class share strong similarities. We clustered these six objects using hierarchical clustering with single linkage. It can be seen that the results obtained with Euclidean distance (Figure 4-*left*) are non-intuitive, linking together objects from different classes. In contrast, the results obtained with the similarity clustering using recurrence plots and CK-1 distance perfectly match the leaf species (Figure 4-*right*).

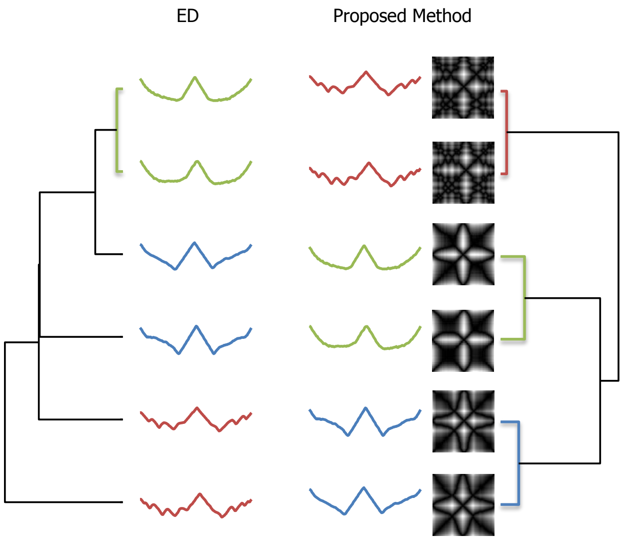

Fig. 4. Dendrograms obtained with six time series from three tree species, using Euclidean distance (*left*) and Recurrence Plots based distance (*right*)

The approach for time series classification using recurrence plots is simple and parameter-free. Algorithm 1 presents the general classification algorithm based on the well-known one nearest neighbor rule as well as the accuracy estimation procedure. Figure 5 illustrates this algorithm.

Algorithm 1 Classification procedure with recurrence plots and CK-1 distance

```
Input: Training data set $S$, testing data set $T$
Output: Estimated accuracy $Acc$
$R \leftarrow \emptyset$
for each $x \in S$ do
    $x' \leftarrow$unthresholded-recurrence-plot($x$)
    $R \leftarrow R \cup \{x'\}$
end for
$Matches \leftarrow 0$
for each $q \in T$ do
    $q' \leftarrow$ unthresholded-recurrence-plot($q$)
    find $r \in R$ that minimizes CK-1($q'$, $r$)
    if the class labels of $r$ and $q$ are the same then
        $Matches \leftarrow Matches + 1$
    end if
end for
$Acc \leftarrow \frac{Matches}{|T|}$
return $Acc$
```

Along the text, we use the term Recurrence Patterns Compression Distance (RPCD) to refer to the CK-1 distance over recurrence plots.

V. Experimental Evaluation

We are very committed with the reproducibility of our results. For this purpose, we created a paper web page [32] in which we made available all detailed numerical results, code and supplemental material not included in this paper. However, we note that our paper is completely self-contained.

We performed a wide experimental evaluation using a large set of time series classification data. In total, the evaluation includes 38 data sets from different domains, such as medicine, entomology, engineering, astronomy, signal processing, and others. More specifically, all the data sets used in our experiments can be found in the UCR Time Series Classification/Clustering Page [20]. These datasets have standard partitions of training and testing, further facilitating the execution of experiments and the comparison of results.

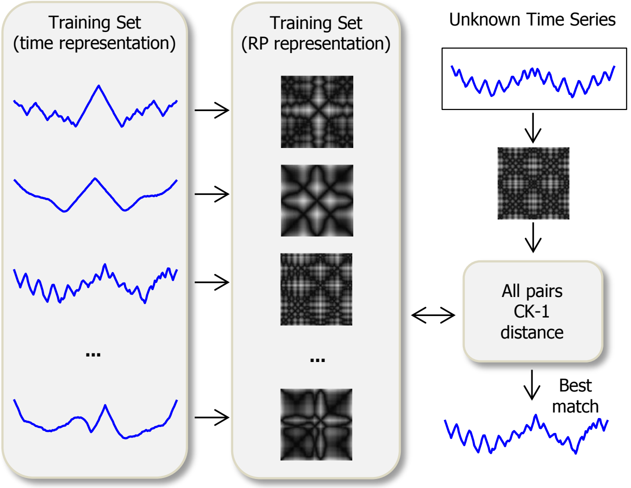

Fig. 5. General procedure to calculate the Recurrence Patterns Compression Distance

The use of benchmark data sets facilitates the reproduction of our results and the direct comparison with other methods proposed in the literature. The UCR time series archive contains 43 data sets, but 5 of them are synthetic since they were generated by some predefined procedure. We decided to exclude all synthetic data sets since they are usually generated for a specific purpose or algorithm. Thus, we ended up with a total of 38 data sets in our experiment.

The experimental section is organized in the following way: we first compare RPCD with the 1-nearest neighbor classifier using Euclidean distance and DTW. We show that our method shows competitive results. More importantly, we show through the Texas Sharpshooter plot that we can predict ahead of time when our method will outperform these state-of-the-art distance measures. Later, we show that CK-1 is indeed the most suitable distance for comparing recurrence plots. We compare CK-1 with Euclidean distance and NCD in the task of classifying these plots.

A. *RPCD against Euclidean and DTW distances*

Several studies have empirically shown that the nearest neighbor algorithm allied to Euclidean or DTW distances provides very accurate classifiers for time series data. Therefore, in this section, we compare the performance of the RPCD against Euclidean and DTW distances for time series in the time domain. The results are reported in Table I (in the column RPCD) together with the accuracy rates obtained by the other approaches.

In order to facilitate the visualization of the results, Figure 6 shows the same results presented in Table I in a graphical representation. In these plots, each data set is represented

TABLE I. ACCURACY RATES FOR EACH DISTANCE MEASURE. THE SYMBOLS ▼, ▲ AND ● REPRESENT DATA SETS GENERATED FROM FIGURE SHAPES, HUMAN MOVEMENTS AND ALL REMAINING DATA SETS, RESPECTIVELY

Data set	**ED**	**DTW**	**RPCD**	**Kind**
50words	63.10	69.00	**77.36**	●
Adiac	61.10	60.40	**61.64**	▼
Beef	53.30	50.00	**63.33**	●
ChlorineConcentration	**65.00**	64.80	51.09	●
CinC ECG torso	89.70	65.10	**97.90**	●
Coffee	75.00	82.10	**100.00**	●
Cricket X	57.40	**77.70**	70.77	▲
Cricket Y	64.40	**79.20**	73.85	▲
Cricket Z	62.00	**79.20**	70.77	▲
DiatomSizeReduction	93.50	**96.70**	96.41	●
ECG200	**88.00**	77.00	86.00	●
ECGFiveDays	79.70	76.80	**86.41**	●
FaceAll	71.40	80.80	**80.95**	▼
FaceFour	78.40	83.00	**94.32**	▼
FacesUCR	76.90	90.49	**94.15**	▼
Fish	78.30	83.30	**87.43**	▼
Gun Point	91.30	90.70	**100.00**	▲
Haptics	37.00	37.70	**38.64**	●
InlineSkate	34.20	**38.40**	32.00	▲
ItalyPowerDemand	**95.50**	95.00	84.26	●
Lighting2	75.40	**86.90**	75.41	●
Lighting7	57.50	**72.60**	64.38	●
MedicalImages	68.40	**73.70**	71.05	●
Motes	**87.90**	83.50	79.71	●
OliveOil	**86.70**	**86.70**	83.33	●
OSULeaf	51.70	59.10	**64.46**	▼
SonyAIBORobotSurface	69.50	72.50	**79.70**	●
SonyAIBORobotSurfaceII	**85.90**	83.10	84.26	●
StarLightCurves	84.90	**90.70**	88.17	●
SwedishLeaf	78.70	79.00	**90.24**	▼
Symbols	90.00	**95.00**	90.45	●
TwoLeadECG	74.70	**90.40**	87.36	●
uWaveGestureLibrary X	**73.90**	72.70	59.30	▲
uWaveGestureLibrary Y	**66.20**	63.40	62.12	▲
uWaveGestureLibrary Z	65.00	**65.80**	61.67	▲
Wafer	99.50	98.00	**99.66**	●
WordsSynonyms	61.80	64.90	**72.41**	●
Yoga	83.00	83.60	**86.60**	▼
Wins	8/38	13/38	**18/38**	

by a point where the y coordinate is the accuracy obtained by comparing RP with CK-1 (RPCD) and the x coordinate represents accuracy obtained by Euclidean distance or DTW. Thus, points above the main diagonal represent data sets in which RPCD outperformed the competing distances.

RPCD is very competitive with Euclidean distance and DTW. Among the 38 data sets used in our experiments, RPCD was superior to Euclidean distance in 28 of them (or 73.68%), and superior to DTW in 20 of them (or 52.63%).

It is interesting to look at the data from different application domains and analyze when RPCD can help. For instance, the UCR archive has data sets generated from figure shapes. It is a representation trick in which shapes of objects (such as leaves and faces) are converted to "time" series by measuring the distance between the object central point and its contour [21]. Figure 7 illustrates this procedure. As we move around the object, we obtain a sequence of distances. Another kind of time series in this repository are those generated by the observation of human movements. Such motions are transformed to time series using accelerometers or by video tracking. To facilitate understanding of the performance of RPCD on these categories, the results shown in Table I and in the graph presented in Figure 6 uses different symbols for each category.

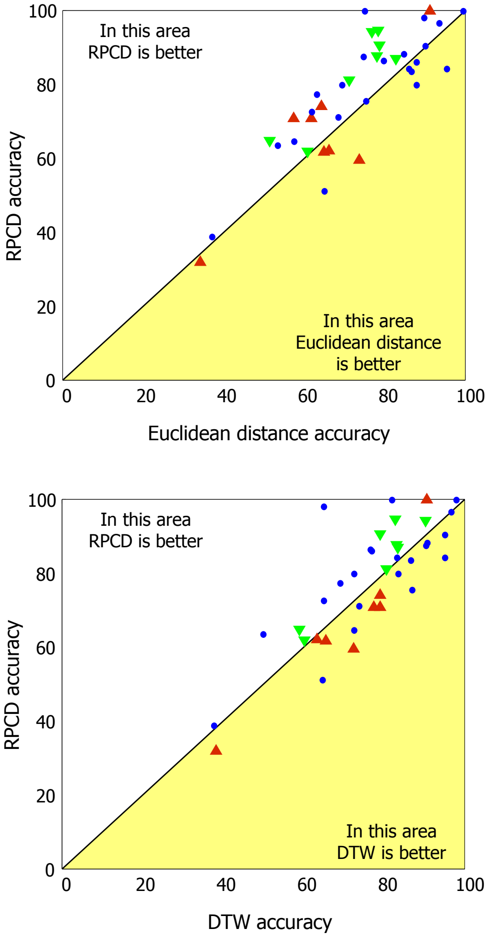

Fig. 6. Graphical representation of the results obtained by RPCD versus Euclidean and DTW distances. Each point represents a different data set. The points above the diagonal represent data sets which the RPCD outperformed Euclidean distance (*top*) or DTW (*bottom*). The symbols ▼, ▲ and ● represent data sets generated from figure shapes, human movements and all remaining data sets, respectively

Observing the results on time series generated from shapes, we can notice that RPCD outperformed Euclidean and DTW distances in all of the eight data sets in this category. We believe that RPCD performs well in this sort of data because shapes frequently result in stationary time series that are characterized by repeating patterns. These patterns may be easily observed in leaf shapes which are frequently cited as examples of natural fractals. Fractals have the property of self-similarity that is characterized by a repeating pattern that occurs in different scales. Other objects that do not have the fractal behavior, such as face contours, may show other properties that also cause repeating patterns such as symmetries and concavities.

As a contrasting example, RPCD does not perform well with time series of human movements. From eight data sets in this category, DTW outperforms RPCD in seven of them. We notice that time series from non-repeating movements may contain too few recurring patterns. Actually, these time series may be highly non-stationary causing that a certain trajectory never visits the same area in the phase space. In contrast, DTW is very effective for this sort of data since warping can deal very well with non-linear time differences between two

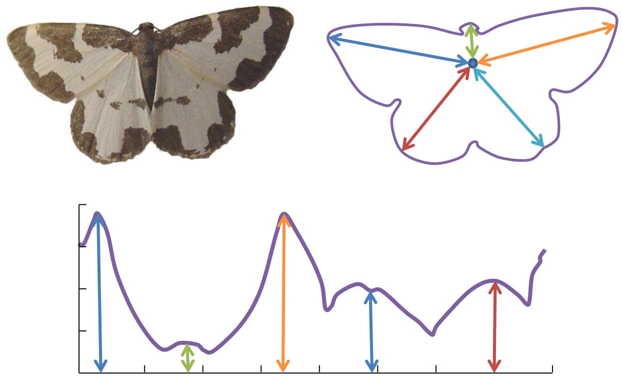

Fig. 7. Example of representation trick in which shapes of objects are converted to "time" series [4]

movements.

We performed a statistical test to detect significant differences among the classification methods. We used the Friedman test with Bonferroni-Dunn post-hoc test [11] at 95% confidence level. The Friedman test rejected the null hypothesis that all methods have similar performance. We proceeded to the Bonferroni-Dunn post-hoc test using the RPCD as control. The post-hoc test indicated a significant difference between RPCD and Euclidean distance, but no significant difference between RPCD and DTW. Figure 8 graphically represents the comparison between the 1-nearest neighbor classifier using RPCD and the same classifier using DTW and ED.

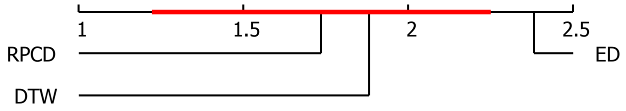

Fig. 8. Graphical representation of the Bonferroni-Dunn post-hoc test. The marked interval represents the range of rankings with no statistical difference in relation to RPCD, used as control

We should interpret the results of the statistical test with a word of caution. First, because our evaluation is quantitative with data sets from different application domains. However, it is very difficult, if not impossible, to propose a time series representation and classification procedure which are the best for all application domains. Nevertheless, RPCD statistically outperformed Euclidean distance and had a similar performance to DTW, which is considered a state-of-the-art distance function for time series classification [12].

As important as having a good average performance on several data sets is to provide a significant improvement in classification accuracy in some relevant problems. We note that RPCD improved considerably the classification performance over both Euclidean distance and DTW in several data sets such as Coffee, ECGFivedays, FaceFour, SwedishLeaf, and others. Obviously, these data sets in which RPCD can significantly improve the classification performance has some sort of match between the data characteristics and the representation bias.

However, RPCD is useless if one cannot identify when it will provide more accurate results. The next section shows how this can be done using a simple plot. The plot indicates that we can safely identify the problems RPCD will outperform other competing methods based on training data performance results.

B. *The Texas Sharpshooter Fallacy*

As we have seen in the previous section, RPCD is able to outperform Euclidean and DTW distances in 73.68% and 52.63% of the data sets, respectively. Although RPCD cannot provide the best accuracy rates for all data sets, and the central assumption of this work is that no single representation is the best for every domain, we can identify when RPCD will outperform the competing methods.

Many papers in the time series classification literature affirm that the proposed method or distance measure is useful since it outperformed the state-of-the-art in some datasets. However, as noted in [5], it is not useful to have an algorithm that can be accurate on some problems unless you can tell in advance on which problems it will be more accurate.

A simple way to show that we can predict when our method will have superior accuracy ahead of time is to use the Texas sharpshooter plot [5]. The Texas sharpshooter fallacy comes from an anecdote of a Texan who fires his gun at the side of a barn, then paints a target around the spot where most bullet holes are clustered [35]. This is in essence the same as evaluating a method that performs well in *some* problems and claiming a posteriori that the method is accurate for the problems it performed well. Such a method has no practical utility unless one can indicate ahead of time for which problems the method will perform well. One way of performing such analysis is testing the accuracy of the competing methods looking only at the training data. We use this information to choose which algorithm will classify the objects from the test set.

In order to do that, we can calculate the accuracy gain, as defined by Equation 10.

$$gain = \frac{accuracy(RPCD)}{accuracy(competition)} \qquad (10)$$

Obviously, gain values greater than one indicate that we expect RPCD will outperform the competition on a given data set; and gain values lower than one indicate the opposite.

Expected gain is the gain calculated over the training set and *actual gain* is the gain over the test set. Recall that the UCR archive provides data sets with standard training and testing splits, and we used these data partitions to calculate the gain values. In order to calculate the gain inside the training set (expected gain) we used leaving-one-out cross-validation, since frequently the training sets have reduced sizes. We also measured the actual gain on testing data, using the accuracy rates presented in Table I. The expected and actual gains for each dataset are presented in Table II.

Figure 9 shows the plots of expected gain versus actual gain. These plots are for RPCD versus Euclidean distance (Figure 9-*top*) and DTW (Figure 9-*bottom*). The plots are divided in four regions:

TABLE II. EXPECTED AND ACTUAL GAINS BETWEEN RPCD AND OTHER DISTANCES

	ED		DTW	
	Expected	Actual	Expected	Actual
50words	1.140	1.226	1.048	1.121
Adiac	1.025	1.009	1.115	1.021
Beef	0.867	1.188	1.300	1.267
ChlorineConcentration	0.797	0.786	0.840	0.788
CinC ECG torso	1.118	1.091	1.357	1.504
Coffee	1.273	1.333	1.273	1.218
Cricket X	1.249	1.233	0.949	0.911
Cricket Y	1.138	1.147	0.886	0.932
Cricket Z	1.263	1.141	0.960	0.894
DiatomSizeReduction	1.000	1.031	1.000	0.997
ECG200	0.988	0.977	1.063	1.117
ECGFiveDays	0.895	1.084	1.000	1.125
FaceAll	1.105	1.134	1.020	1.002
FaceFour	1.250	1.203	1.000	1.136
FacesUCR	1.245	1.224	0.984	1.040
Fish	1.113	1.117	1.138	1.050
Gun Point	1.042	1.095	1.250	1.103
Haptics	0.975	1.044	1.258	1.025
InlineSkate	1.000	0.936	0.714	0.833
ItalyPowerDemand	0.922	0.882	0.922	0.887
Lighting2	1.093	1.000	0.959	0.868
Lighting7	1.044	1.120	0.979	0.887
MedicalImages	0.993	1.039	0.954	0.964
Motes	0.867	0.907	0.867	0.955
OliveOil	1.000	0.961	1.040	0.961
OSULeaf	1.169	1.247	1.014	1.091
SonyAIBORobotSurface	0.947	1.147	1.002	1.099
SonyAIBORobotSurfaceII	1.057	0.981	1.057	1.014
StarLightCurves	1.036	1.039	1.001	0.972
SwedishLeaf	1.197	1.147	1.182	1.142
Symbols	1.048	1.005	0.917	0.952
TwoLeadECG	1.222	1.169	1.048	0.966
uWaveGestureLibrary X	0.787	0.802	0.879	0.816
uWaveGestureLibrary Y	0.905	0.938	0.884	0.980
uWaveGestureLibrary Z	0.919	0.949	0.882	0.937
Wafer	1.002	1.002	1.004	1.017
WordsSynonyms	1.191	1.172	1.051	1.116
Yoga	1.095	1.043	1.045	1.036

TP In this region we claimed ahead of time that RPCD would improve accuracy, and we were correct;

TN In this region we correctly claimed ahead of time that RPCD would decrease accuracy;

FN In this region we claimed ahead of time that RPCD would decrease accuracy, but the accuracy actually increased. This represents a lost opportunity to improve, but note that we are no worse off than if we had not tried RPCD;

FP This region is the only truly bad case for our method. Data points falling in this region represent cases where we thought we could improve accuracy, but did not.

Most of the points on the plots of Figure 9 fall in the TP and TN areas. This means that we can confidently predict that RPCD will outperform or will be outperformed by the competing methods. Concretely, for Euclidean distance, 31 points fall in these areas (or 81.58%); and for DTW, 34 points (or 89,47%).

There are only 6 points falling in the FN area of the two plots (5 for ED and 1 for DTW). These points indicate an incorrect diagnosis, i.e., we predicted a performance loss for RPCD, but it actually improved classification accuracy. However, we can understand this situation as a lost opportunity, since we reject to use RPCD, but we are no worse off than if we had not tried RPCD.

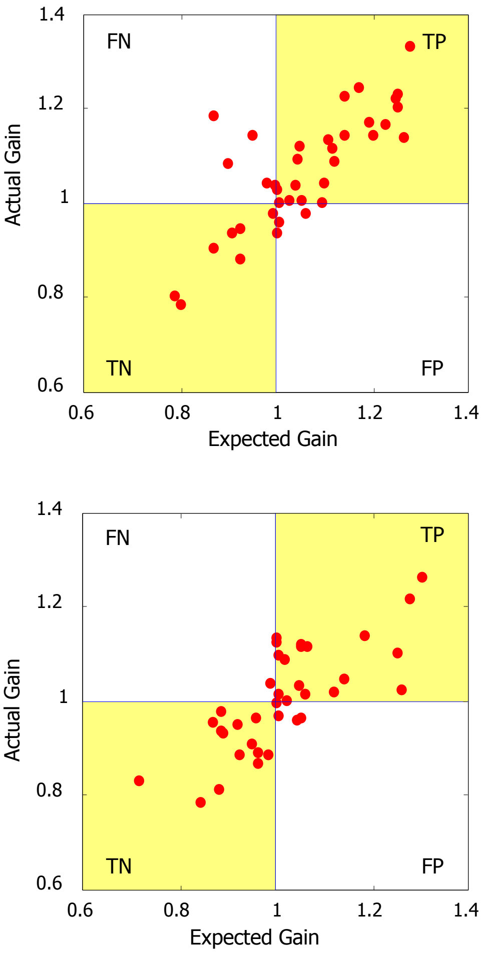

Fig. 9. Texas sharpshooter plot for RPCD versus ED (*top*) and DTW (*bottom*)

The bad case for RPCD occurs in the FP area. However, there are only 5 points fall in this area, all of them representing minor gains or losses that could easily happen by chance.

C. *Distances Between Recurrence Plots*

At this point, we have shown that RPCD can be an accurate approach for classifying time series. However, it is interesting to analyze where the accuracy of RPCD comes from. In particular, we claimed that CK-1 is a suitable distance for recurrence plots, since it can make use of the spatial information of the images. In this section, we show that CK-1 can indeed outperform other distance measures between recurrence plots for time series classification. We performed an experiment using the 1-nearest neighbor algorithm with CK-1, Euclidean and the Normalized Compression distances, such as used in [6]. The NCD was calculated using the CompLearn NCD implementation [1]. The results are presented in Table III.

Figure 10 shows the same results as presented in Table III in a graphical representation. In these plots, each data set is represented by a point where the y coordinate is the accuracy obtained by comparing RP with CK-1 (RPCD) and the x coordinate represents accuracy obtained by Euclidean distance

[1] http://complearn.org/ncd.html

TABLE III. ACCURACY RATES FOR 1-NEAREST NEIGHBOR CLASSIFIER WITH ED, NCD AND CK-1 (RPCD) DISTANCES OF RECURRENCE PLOTS

Data set	ED	NCD	RPCD
50words	63.08	5.93	**77.36**
Adiac	**62.92**	54.22	61.64
Beef	**66.67**	23.33	63.33
ChlorineConcentration	**63.70**	62.32	51.09
CinC ECG torso	97.25	51.52	**97.90**
Coffee	**100.00**	53.57	**100.00**
Cricket X	53.85	12.31	**70.77**
Cricket Y	54.87	8.21	**73.85**
Cricket Z	56.15	12.05	**70.77**
DiatomSizeReduction	96.08	69.28	**96.41**
ECG200	**87.00**	71.00	86.00
ECGFiveDays	80.49	76.07	**86.41**
FaceAll	74.38	2.01	**80.95**
FaceFour	79.55	25.00	**94.32**
FacesUCR	75.46	17.71	**94.15**
Fish	78.29	37.14	**87.43**
Gun Point	89.33	94.00	**100.00**
Haptics	**39.61**	23.38	38.64
InlineSkate	**33.09**	22.18	32.00
ItalyPowerDemand	**96.21**	61.61	84.26
Lighting2	73.77	52.46	**75.41**
Lighting7	54.79	21.92	**64.38**
MedicalImages	69.08	57.89	**71.05**
Motes	**79.87**	62.78	79.71
OliveOil	**86.67**	26.67	83.33
OSULeaf	52.89	48.35	**64.46**
SonyAIBORobotSurface	76.71	57.07	**79.70**
SonyAIBORobotSurfaceII	**85.10**	62.96	84.26
StarLightCurves	**88.96**	75.11	88.17
SwedishLeaf	82.40	44.16	**90.24**
Symbols	87.74	49.85	**90.45**
TwoLeadECG	75.68	**94.38**	87.36
uWaveGestureLibrary X	56.39	30.82	**59.30**
uWaveGestureLibrary Y	59.55	24.96	**62.12**
uWaveGestureLibrary Z	58.24	30.63	**61.67**
Wafer	99.58	99.53	**99.66**
WordsSynonyms	61.29	15.05	**72.41**
Yoga	82.37	66.90	**86.60**
Wins	12/38	1/38	**26/38**

or NCD. Thus, points above the main diagonal represent data sets in which RPCD outperformed the competing distance.

From the results, it is very evident that CK-1 outperforms Euclidean distance and NCD. In particular, RPCD outperformed NCD in all but two datasets, in many of them by a very large margin. Nevertheless, we performed a hypothesis test, in order to verify if the performance of the compared methods are statistically different. Since NCD performed poorly in our experiment, we excluded it from the hypothesis test. We used the Friedman test with the Bonferroni-Dunn post-hoc test at 95% confidence level. The test rejected the null hypothesis that RPCD and Euclidean distance have similar performances. Figure 11 graphically represents the comparison between the 1-nearest neighbor classifier using RPCD and ED.

VI. CONCLUSION

In this work we propose the use of recurrence plots as representation domain for time series classification. Our approach employs similarity-based classification using the Campana-Keogh (CK-1) distance. Our method is a simple and parameter-free approach to time series classification.

To ensure the effectiveness of the proposed method, we conducted experiments on a large number of data sets. The results show that the proposed algorithm outperforms Euclidean distance in 73.68% and DTW in 52.63% of the data sets. In some data sets, RPCD outperforms the competing distances by a large margin. We also showed with the Texas sharpshooter plot that we can predict ahead of time in which data sets RPCD will outperform the other distances.

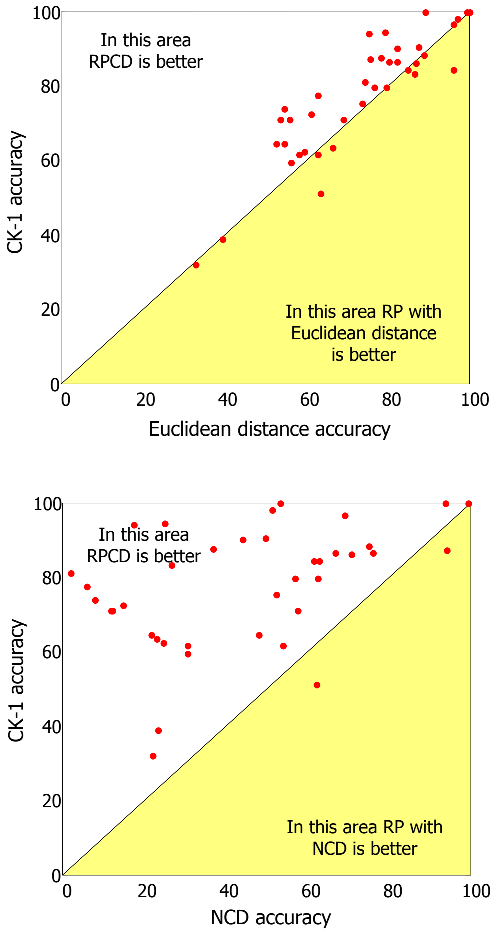

Fig. 10. Graphical representation of the results obtained by CK-1, Euclidean Distance and NCD over RP. Each point represents a different data set. The points above the diagonal represent data sets which the RP compared with CK-1 outperformed Euclidean distance (*top*) or NCD (*bottom*)

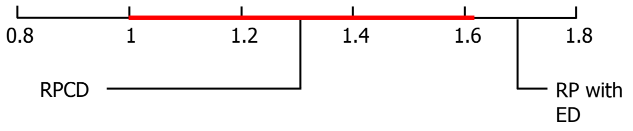

Fig. 11. Graphical representation of the Bonferroni-Dunn post-hoc test. The marked interval represents the range of rankings with no statistical difference in relation to RPCD, used as control

As future work we indent to evaluate the influence of different time series characteristics, such as non-stationarity, on the performance of the RPCD in classification. We will look for different data pre-processing techniques to remove such characteristics from the data, if necessary. We will also investigate the use of RPCD in other data mining tasks such as clustering, anomaly detection and motif discovery. Finally, we will evaluate the robustness of our method to provide invariances in the time series, such as rotation, phase and scale.

Acknowledgment

This work was funded by São Paulo Research Foundation (FAPESP). Grants #2011/04054-2, #2011/17698-5, #2012/07295-3 and #2012/50714-7.

References

[1] A. Bagnall, L. M. Davis, J. Hills, and J. Lines, "Transformation based ensembles for time series classification," in *Proceedings of the 12th SIAM International Conference on Data Mining*, 2012, pp. 307–318.

[2] J. A. Bastos and J. Caiado, "Recurrence quantification analysis of global stock markets," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 7, pp. 1315–1325, 2011.

[3] G. E. A. P. A. Batista, E. J. Keogh, A. Mafra-Neto, and E. Rowton, "Sensors and software to allow computational entomology, an emerging application of data mining," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011, pp. 761–764.

[4] G. E. A. P. A. Batista, B. J. L. Campana, and E. J. Keogh, "Classification of live moths combining texture, color and shape primitives," in *Proceedings of the 9th International Conference on Machine Learning and Applications*, 2010, pp. 903–906.

[5] G. E. A. P. A. Batista, X. Wang, and E. J. Keogh, "A complexity-invariant distance measure for time series," in *Proceedings of the 11th SIAM International Conference on Data Mining*, 2011, pp. 699–710.

[6] J. P. Bello, "Measuring structural similarity in music," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 19, no. 7, pp. 2013–2025, 2011.

[7] A. Bratko, G. Cormack, B. Filipic, T. Lynam, and B. Zupan, "Spam filtering using statistical data compression models," *Journal of Machine Learning Research*, vol. 7, pp. 2673–2698, 2006.

[8] B. J. L. Campana and E. J. Keogh, "A compression based distance measure for texture," in *Proceedings of the 10th SIAM International Conference on Data Mining*, 2010, pp. 850–861.

[9] S. Carrubba, A. Minagar, A. Chesson Jr, C. Frilot, and A. Marino, "Increased determinism in brain electrical activity occurs in association with multiple sclerosis," *Neurological Research*, vol. 34, no. 3, pp. 286–290, 2012.

[10] E. Delaherche, M. Chetouani, A. Mahdhaoui, C. Saint-Georges, S. Viaux, and D. Cohen, "Interpersonal synchrony: A survey of evaluation methods across disciplines," *IEEE Transactions on Affective Computing*, vol. 3, no. 3, pp. 349–365, 2012.

[11] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[12] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proceedings of the 34th International Conference on Very Large Data Bases Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[13] S. Dua, X. Du, S. V. Sree, and T. A. V. I., "Novel classification of coronary artery disease using heart rate variability analysis," *Journal of Mechanics in Medicine and Biology*, vol. 12, no. 4, pp. 1–19, 2012.

[14] J. P. Eckmann, O. S. Kamphorst, and D. Ruelle, "Recurrence plots of dynamical systems," *Europhysics Letters*, vol. 4, no. 9, pp. 973–977, 1987.

[15] M. González-Izal, A. Malanda, E. Gorostiaga, and M. Izquierdo, "Electromyographic models to assess muscle fatigue," *Journal of Electromyography and Kinesiology*, vol. 22, no. 4, pp. 501–512, 2012.

[16] Y. Hao, B. J. L. Campana, and E. J. Keogh, "Monitoring and mining insect sounds in visual space." in *Proceedings of the 12th SIAM Conference on Data Mining*, 2012, pp. 792–803.

[17] B. Hu, T. Rakthanmanon, B. J. L. Campana, A. Mueen, and E. J. Keogh, "Image mining of historical manuscripts to establish provenance." in *Proceedings of the 12th SIAM Conference on Data Mining*, 2012, pp. 804–815.

[18] J. S. Iwanski and E. Bradley, "Recurrence plots of experimental data: To embed or not to embed?" *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 8, no. 4, pp. 861–871, 1998.

[19] E. J. Keogh, S. Lonardi, C. A. Ratanamahatana, L. Wei, S. Lee, and J. Handley, "Compression-based data mining of sequential data," *Data Mining and Knowledge Discovery*, vol. 14, no. 1, pp. 99–129, 2007.

[20] E. J. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: a survey and empirical demonstration," in *Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 102–111.

[21] E. J. Keogh, L. Wei, X. Xi, M. Vlachos, S.-H. Lee, and P. Protopapas, "Supporting exact indexing of arbitrarily rotated shapes and periodic time series under euclidean and warping distance measures," *VLDB Journal*, vol. 18, no. 3, pp. 611–630, 2009.

[22] N. Krasnogor and D. A. Pelta, "Measuring the similarity of protein structures by means of the universal similarity metric," *Bioinformatics*, vol. 20, pp. 1015–1021, 2004.

[23] M. Li, J. H. Badger, X. Chen, S. Kwong, P. Kearney, and H. Zhang, "An information-based sequence distance and its application to whole mitochondrial genome phylogeny," *Bioinformatics*, vol. 17, no. 2, pp. 149–154, 2001.

[24] M. Li and P. Vitanyi, *An introduction to Kolmogorov complexity and its applications*, 2nd ed. Springer Verlag, 1997.

[25] M. Li, X. Chen, X. Li, B. Ma, and P. Vitanyi, "The similarity metric," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3250–3264, 2004.

[26] J. Lines, L. M. Davis, J. Hills, and A. Bagnall, "A shapelet transform for time series classification," in *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 289–297.

[27] N. Marwan, M. C. Romano, M. Thiel, and J. Kurths, "Recurrence plots for the analysis of complex systems," *Physics Reports*, vol. 438, no. 5Ð6, pp. 237 – 329, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0370157306004066

[28] A. T. Monk and A. H. Compton, "Recurrence phenomena in cosmic ray intensity," *Reviews of Modern Physics*, vol. 11, no. 3–4, pp. 173–179, 1939.

[29] F. M. Neves, M. R. Pie, and R. L. Viana, "Self-organization in the movement activity of social insects (hymenoptera: Formicidae)," in *Proceedings of the 10th International Conference of Numerical Analysis and Applied Mathematics*, vol. 1479, no. 1, 2012, pp. 658–661.

[30] G. Ouyang, Z. Ju, and H. Liu, "Surface EMG signals determinism analysis based on recurrence plot for hand grasps," in *Proceedings of the 2012 International Joint Conference on Neural Networks*, 2012, pp. 1–6.

[31] G. Shao and Y. Chen, "Prediction protein structural classes with a hybrid feature," in *Proceedings of the 2012 IEEE Symposium on Electrical Electronics Engineering*, 2012, pp. 202–205.

[32] D. F. Silva, V. M. A. Souza, and G. E. A. P. A. Batista, "Website to this work," http://sites.labic.icmc.usp.br/dfs/rpcd/.

[33] D. F. Silva, V. M. Souza, G. E. Batista, and R. Giusti, "Spoken digit recognition in portuguese using line spectral frequencies," in *Advances in Artificial Intelligence – IBERAMIA 2012*, ser. Lecture Notes in Computer Science, J. Pavan, N. Duque-Mendez, and R. Fuentes-Fernandez, Eds. Springer Berlin Heidelberg, 2012, vol. 7637, pp. 241–250.

[34] D. P. Subha, P. K. Joseph, R. Acharya U, and C. M. M. Lim, "EEG signal analysis: a survey," *Journal of Medical Systems*, vol. 34, no. 2, pp. 195–212, 2010.

[35] W. C. Thompson, "Painting the target around the matching profile: the Texas sharpshooter fallacy in forensic DNA interpretation," *Law, Probability and Risk*, vol. 8, no. 3, pp. 257–276, 2009.

[36] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. J. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 275–309, 2013.