A Software Tool for the Analysis of Similarity in Recurrence Patterns

Ernesto Bautista-Thompson[1], Roberto Brito-Guevara[1], Jesús E. Molinar-Solis[2]

1Centro de Tecnologías de la Información, DES-DACI, Universidad Autónoma del Carmen, Calle 56 Número 4, C. P. 24180, Ciudad del Carmen, Campeche, México
[2]Centro Universitario UAEM-Ecatepec, Universidad Autónoma del Estado de México José Revueltas 17, Col. Tierra Blanca, C. P. 55020, Ecatepec de Morelos, Estado de México, México
teb_thompson@yahoo.com, rbritoguevara@hotmail.com, molinarov@hotmail.com

Abstract. Recurrence plots visualize spatial and also in the case of time series temporal correlations inside sequences of data, the technique allows the identification of hidden data relationships (periodicity, non-stationarity, recurrence, randomness) inside a sequence. The comparison between recurrence patterns in order to identify common structures is difficult because the lack of similarity quantification tools in the available and most popular software for recurrence plots analysis such as VRA (Visual Recurrence Analysis), RQA (Recurrence Quantification Analysis) and CRP (Cross Recurrence Plots). In this work a software tool for analysis of structural similarity patterns between recurrence plots is proposed, this tool named RecurrenceVs, allows the comparison and quantification of the degree of structural similarity between recurrence plots generated from different sequences of data. The results shows that this tool is useful for the classification of data sequences by similarity families based on the recurrence patterns, where these patterns preserve the information about the structure and dynamics of data sequences.

Keywords: Recurrence Patterns, Spatial and Temporal Correlation, Structural Similarity.

1 Introduction

The extraction of common structural features in sets of time series and sequences of data is an important task in the identification of patterns of interest for example: in the analysis of time series dynamics [1], the identification of motifs in genomics sequences [2], the construction of queries for sequence extraction in databases [3]. Similarity in data sequences can be measure with different metrics such as Euclidean distance [4], Dynamic Time Warping [5], Similarity Histograms [6], etc. such measures operates directly over the data sequences. For other side, analysis of the dynamics of sequences of data, understood as the relationships between the different data inside a sequence, can be done with different techniques such as: Autocorrelation [7], Wavelet Analysis [8], Recurrence Quantification Analysis [9], etc. In particular, the

Advances in Computer Science and Engineering.
Research Computing Science 45, 2010, pp. 109-118
Received 30/04/10
Final version 19/05/10
Accepted 14/05/10

Recurrence Quantification Analysis is based on the generation of a data representation named recurrence plots, these plots shows patterns of hidden relationships between data such as non-stationary behavior, periodicities, and randomness; the patterns generated with these plots can be used to compare sequences of data at a new level of information. There are not quantitative tools for comparison of recurrence patterns, software such as VRA, RQA and CRP lacks this functionality [10, 11, 12], this is the main motivation for the development and proposal in this work of a software tool for quantitative comparison and analysis of similarity between recurrence patterns. Section two, explains the theoretical basis of the recurrence plots as well as some concepts of similarity. Section three, describes the technical aspects of the RecurrenceVs software tool. In Section fourth, the experimental results of the evaluation of the software tool are presented. Finally in section five, the discussion of this work is presented.

2 Recurrence Plots

Recurrence plots are graphical representation of a sequence of data, which allows the detection of hidden dynamical patterns and nonlinearities inside the data. These plots allow the visualization of recurrent patterns, non-stationary patterns, and structural changes. The recurrence plot is part of a technique known as Recurrence Analysis that was developed by Eckman et al. in 1987 [9].

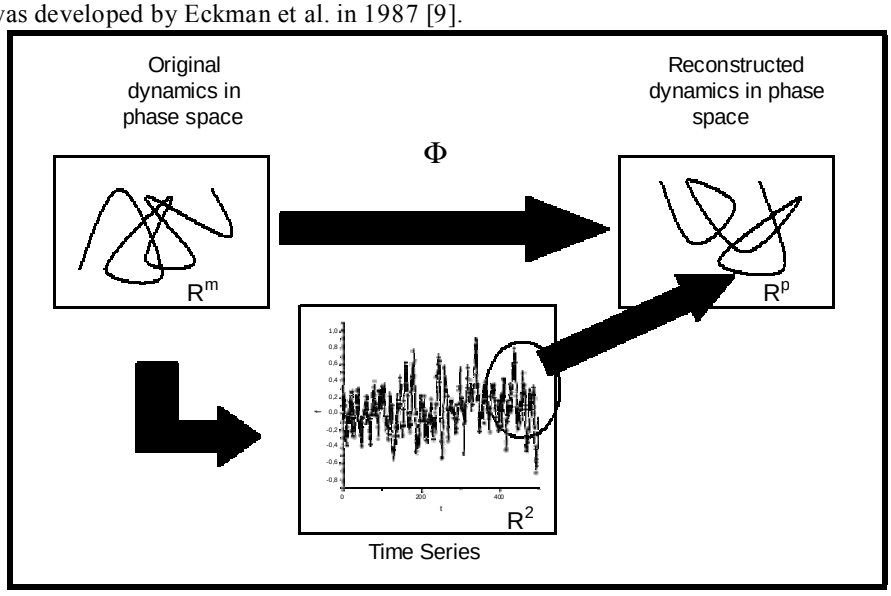

Fig. 1. Conceptual schema of reconstruction of a multidimensional system based on a observable (time series).

The theoretical foundation of the construction of recurrence plots is based on two theorems proposed by H. Whitney and F. Takens [13, 14], in these foundational works they establish that is possible to recreate a topological equivalent of the original

behavior from a multidimensional system by means of a sequence of data from an observable of such system [14], the Fig.1 illustrates this idea.

In the general case of the reconstruction of a system with embedding dimension *n* and a delay *t=0,* each datum *x(i)* is a vector composed by *n* consecutive data elements from the sequence.

$$x(i) = \{x(1), x(2), x(3), ..., x(n)\}. \quad (1)$$

A recurrence plot is generated by comparison of each datum in a sequence with itself and with the rest of data. The comparison between, for example, data *x(i)* and *x(j)* is made with a metric such as Euclidean distance.

$$d_{ij} = \|x(i) - x(j)\|. \quad (2)$$

This allows building a correlation matrix *D* of spatial and temporal nature (as is the case of time series).

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{bmatrix}. \quad (3)$$

Each element d_{ij} in matrix D is associated with the Euclidean distance between a datum in position *i* and a datum in position *j* inside a sequence, if a datum o subsets of data are recurrent this behavior will be detected by means of sets of equal distances. Also, each distance d_{ij} is associated with a value from a discretized gray tone scale, in this way an image of the recurrence plot pattern is generated. The resultant pattern is symmetric due to the redundancy in the calculation of the distances $d_{ij} = d_{ji}$.

Comparing the patterns generated for the recurrence plots can be useful in order to identify sequences of data with similar data behavior or dynamics, this similarity must be quantified in order to have an objective comparison of the recurrence plots.

3 Design of RecurrenceVs: Architecture and Algorithms

The design of the software tool has two important aspects: user interface design and algorithm design. In the first case, the user interface must be easy of use and to be windows driven, the software must allow the execution of a series of multiple experiments and it must be easy to save the results in incremental way. The user interface features were motivated by the analysis of different software tools for recurrence plot analysis (VRA, RQA, CRP), in Table 1, a comparative feature analysis between these

software tools is showed and in Fig. 2 a screenshot of the RecurrenceVs interface is presented.

Table 1. Comparison of features between different tools.

Comparison of Features for Recurrence Analysis Software			
Software	User Inteface	Similarity Analysis Function	Multiple Experiments in one Run
RQA	DOS Command Line	No	Yes, in Batch Mode
VRA	GUI	No	No
CRP	MatLab Command Line	No	Needs Programming
RecurrenceVs	GUI	Yes	Yes, in User Sesion

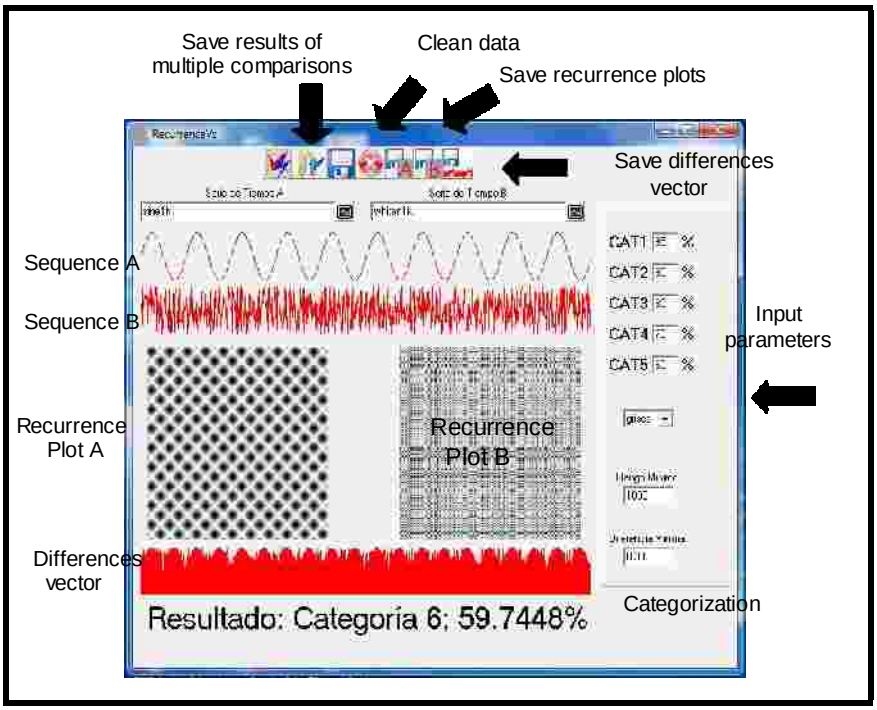

Fig. 2. Screenshot of the software interface and its elements.

The design of the algorithms was divided in three stages: first the load and preparation of the data, second the construction and comparison of the recurrence plots and third the visualization and storage of the results. In the first stage, an important aspect if the normalization of the data sequences to be compared, the data sequences can

have different length, and data range (the data types used are: integer and real), in order to generate comparable recurrence plots they are normalized to a default length of 1000 data by sampling each sequence, but this parameter of length can be adjusted if shorter time series are used in the similarity analysis. The normalization in range is done by scaling the values in order to have a range of values between 0 to 1, the Fig. 3 shows a diagram of this stage.

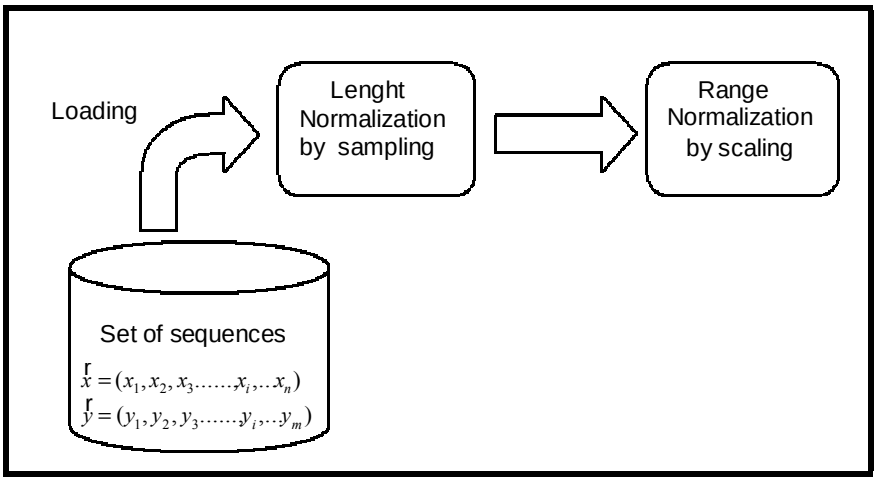

Fig. 3. Stage one, preprocessing.

The construction of the recurrence plots is done with the algorithm described in Section two, the comparison between the recurrence plots is done by comparing the corresponding element $[i, j, d_{ij}]$ of the matrix of distances for each of the two recurrence plots, then for each position *(i, j)* the difference between the corresponding distances d_{ij} for both recurrence plots is calculated, a difference of distances $D_{d_{ij}}$ is considered as a match between both recurrence plots if it has a value below a similarity threshold S_t, this input parameter is setup by the user before the beginning of the analysis, the threshold allows to establish different degrees of similarity between the recurrence plots, the values of the differences between the distances d_{ij} are discretized as 1 for those bellow the threshold and 0 above the threshold and storage in a distance vector, a counting of the number of minimum differences is done, in order to determine the percentage of minimum distances between the compared recurrence plots, a set of five similarity percentages are established by the user, and depending on the corresponding percentage reached by the counting process a similarity degree between the recurrence plots is determined. The Fig. 4 shows the steps of this second stage.

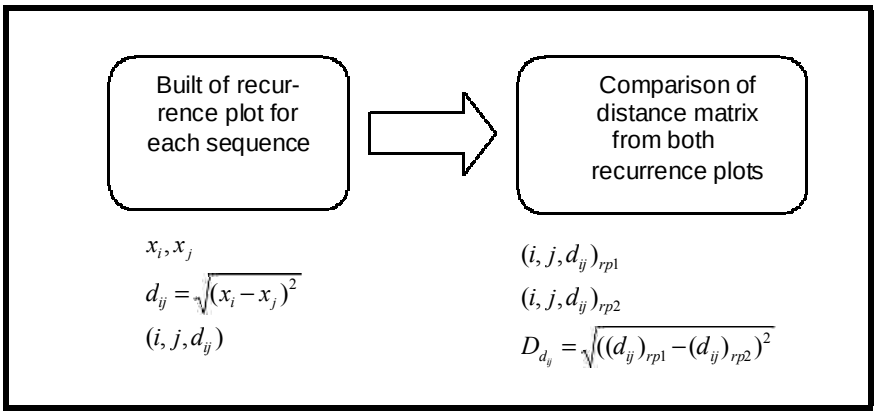

Fig. 4. Stage two, recurrence plot comparison.

Finally, in the third stage the visualization of the recurrence plots and their results are showed, the recurrence plots are visualized using a gray scale that corresponds with the distance between their points, the white color represents the minimum distance and the black color corresponds to the maximum distance present in the plot, each graphic of a recurrence plots is generated with a sampling of their corresponding distance set in order to facilitate its visualization, this is due to the enormous quantity of data generated, for example for a sequence of 1000 data a matrix of 10^6 points is generated. In this stage, the option to save the recurrence plots and the results of their comparison is activated, multiple comparison experiments can be done in one run for example: comparing a specific recurrence plot against a set of different recurrence plots can be saved by incremental storage of each new experiment, in the Fig. 5 is showed the steps of this last stage.

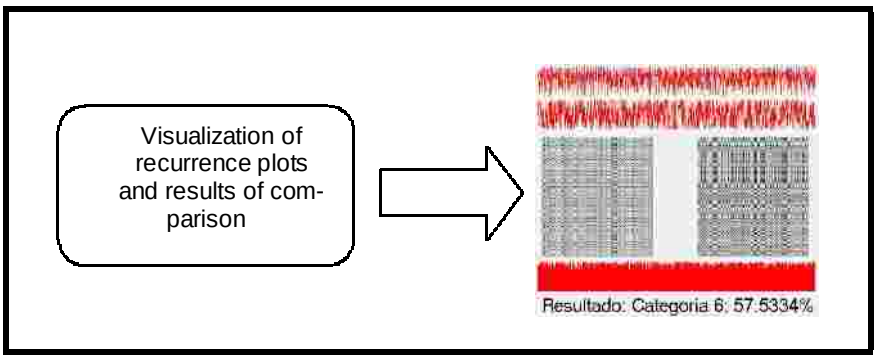

Fig. 5. Stage three, visualization of the results.

4 Evaluation: Performance and Usability

The software tool was evaluated with a set of synthetic data sequences previously classified by their similarity degree by means of the direct comparison of the sequences using the technique of Derivative Dynamic Time Warping (DDTW) [15, 16]. Examples of different comparisons are showed in the screenshots of Fig. 6 and Fig. 7; in these examples a recurrence plot from a sequence tagged DS11 is compared with the corresponding recurrence plots for the sequences DS12 and DS42; in the previous classification DDTW reported in [15, 16], the similarity of these sequences groups DS11 and DS12 in the same class 5 and the sequence DS42 belongs to class 1. The results with RecurrenceVs show a correspondence with the aforementioned results.

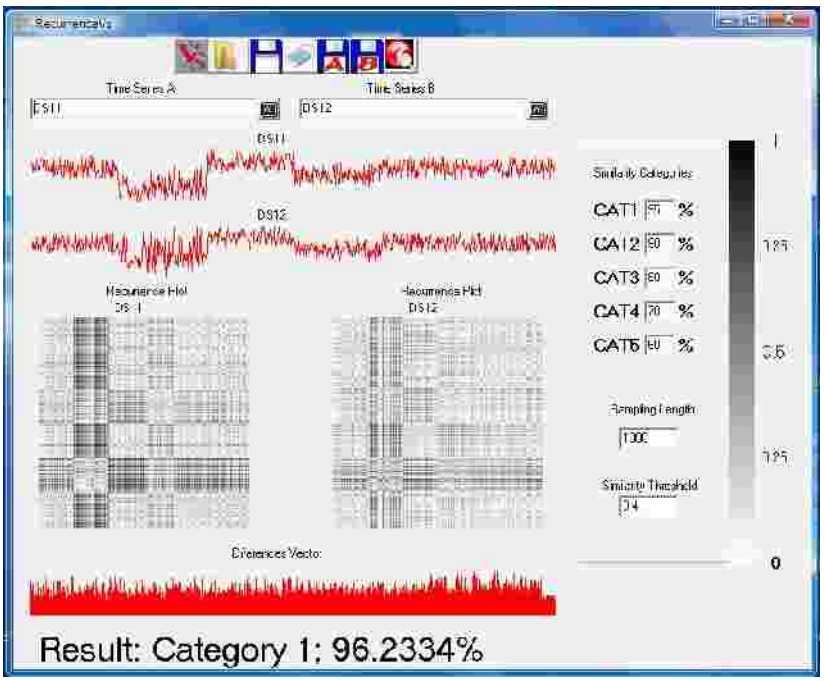

Fig. 6. Screenshot of the similarity analysis for the recurrence plots of two synthetic data sequences DS11 and DS12, their similarity corresponds to category 1 (95% of similarity).

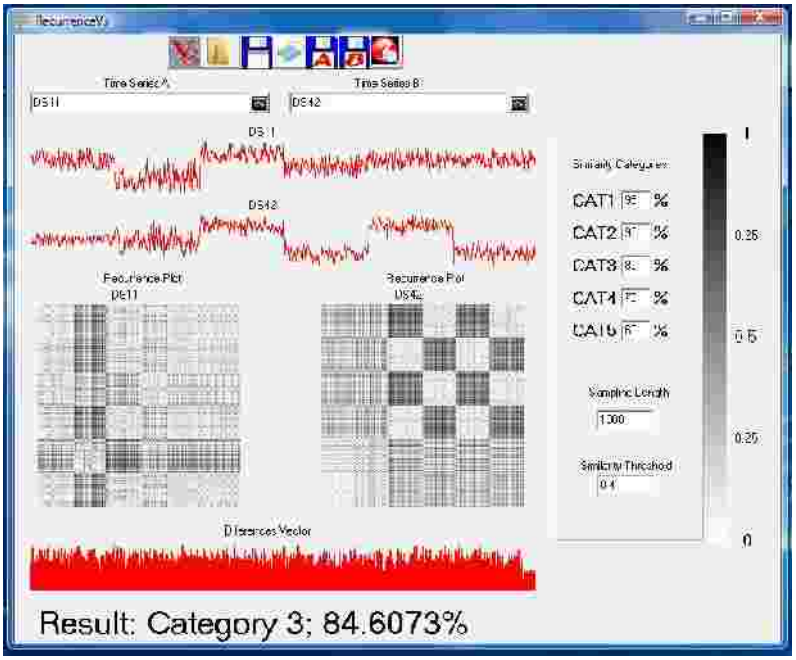

Fig. 7. Screenshot of the similarity analysis for the recurrence plots of two synthetic data sequences DS11 and DS42, their similarity corresponds to category 3 (84.6% of similarity).

In Fig. 8 is show, the behavior of the similarity percentage for the similarity analysis made with the RecurrenceVs tool, for the sequence DS12 that belongs to the class 5 in the DDTW classification [15, 16].

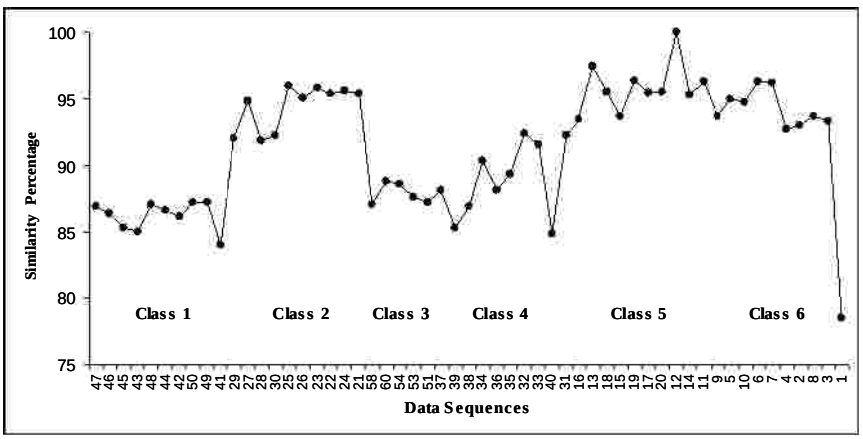

Fig. 8. Behavior of the similarity percentage between the recurrence plots for the data sequence DS12 and the rest of the sequences from the experimental set.

The sensibility of the input parameters is illustrated in Table 2, it shows the effect that changes in the similarity threshold S_t have on the categorization of the recurrence

plots, in this example for the sequences DS11 and DS12, the sensibility is greater for a value of S_t= 0.1, this corresponds with a requirement of almost identical recurrent plots, for values of S_t between 0.3 and 0.4 the sensibility is reduced and no change in the categorization is observed.

Table 2. Sensibility of input parameters, categorization of the compared recurrence plots for the DS11 and DS12 similar data sequences.

Sensibility of input parameters Comparison of two similar sequences			
Similarity threshold	Assigned category	Percentage of similarity	Threshold of similarity percentage
0.1	6	57.28%	below 60%
0.2	3	86.55%	80%
0.3	1	97.17%	95%
0.4	1	99.59%	95%
0.1	5	57.28%	50%
0.2	2	86.55%	80%
0.3	1	97.17%	90%
0.4	1	99.59%	90%

5 Discussion

A software tool for the analysis of similarity between recurrence patterns was developed, this tool identified as RecurrenceVs includes a user-friendly interface in order to develop a series of experiments for the study of similarity between sets of recurrence plots where these represents different dynamical behaviors from sequences of data. The evaluation of the software tool with the set of time series from [15, 16] shows that it is capable of discriminate between similar and non similar sequences based on their recurrence representations and generate classifications based on the recurrence patterns. The parameterization of the similarity by means of a threshold and a similarity percentage is useful because it allows the analysis of similarity between recurrence patterns where their differences are not sharp. This analysis tool can be a complement to the existing recurrence analysis tools (VRA, RQA, CRP) where different properties such as percent of recurrence, percent of determinism, Shannon entropy, etc. can be calculated for each recurrence plot and in this way correlate the similar recurrence patterns with such properties.

References

1. Bautista-Thompson, E.: Measurement of Time Series Predictability: An Experimental Study, Ph. D. Thesis. CIC-IPN, México D.F. (2005).
2. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195--197 (1981)
3. Das, G., Gunopulos, D., Mannila H.: Finding Similar Time Series. In: Komorowski, H.J., Zytkow, J.M. (eds.) PKDD'97, LNCS, vol. 1263, pp. 88-100. Springer-Verlag, London (1997).
4. Yi, B.K., Faloutsos, C.: Fast Time Sequence Indexing for Arbitrary Lp Norms. In: Proceedings of the 26th International Conference on Very large Databases, pp. 385-394. Morgan Kauffmann, San Francisco (2000).
5. Keogh, E., Pazzani, M.: Scaling Up Dynamic Time Warping for Data Mining Applications. In: Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 285-289. ACM Press, Boston (2000).
6. Bautista-Thompson, E., Santos-De la Cruz, S.: Shape Similarity Index for Time Series Based on Features of Euclidean Distances Histograms. In: Gelbuck, A., Suárez-Guerra, S. (eds.) Proceedings of the 15th International Conference on Computing, pp. 60-64. IEEE Computer Society Press, Los Alamitos (2006).
7. Proakis, J. G., Manolakis, D. K.: Digital Signal Processing Principles, Algorithms, and Applications. Prentice Hall (2006)
8. Mallat, S.: A Wavelet Tour of Signal Processing: The Sparce Way. Academic Press, Burlington (2009)
9. Eckmann, J.P., Kampshort, S.O., Ruelle, D.: Recurrence Plots of Dynamical Systems. Europhys. Lett. 4, 973-977 (1987).
10. Visual Recurrence Analysis Software (by Eugene Kononov), http:// www.myjavaserver.com/~nonlinear/vra/download.html
11. Zbilut, J.P., Weber, C.L.: Embeddings and Delays as Derived from Quantification of Recurrence Plots. Phys. Lett. A 171, 199-203 (1992).
12. Marwan, N.: Encounters with Neighbours: Current Developments of Concepts Based on Recurrence Plots and Their Applications, Ph. D. Thesis. Potsdam University, Potsdam (2003).
13. Whitney, H.: Differentiable Manifolds. Annals of Mathematics 37, 645-680 (1934).
14. Takens, F.: Detecting Strange Attractors in Turbulence. Lecture Notes in Mathematics 898, 366-381 (1981).
15. Keogh, E., Pazzani, M.: Scaling Up Dynamic Time Warping for Data Mining Applications. In: Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 285-289. ACM Press, Boston (2000).
16. Hettich, S., Bay, S. D.: The UCI KDD Archive, http://kdd.ics.uci.edu.