Physics Letters A 379 (2015) 997–1011

Contents lists available at ScienceDirect

Physics Letters A

www.elsevier.com/locate/pla

Approximation of diagonal line based measures in recurrence quantification analysis

David Schultz [a,*], Stephan Spiegel [a], Norbert Marwan [b], Sahin Albayrak [a]

[a] DAI-Lab, Berlin Institute of Technology, Ernst-Reuter-Platz 7, 10587 Berlin, Germany
[b] Potsdam Institute for Climate Impact Research, 14412 Potsdam, Germany

ARTICLE INFO

Article history:
Received 25 November 2014
Received in revised form 24 January 2015
Accepted 27 January 2015
Available online 29 January 2015
Communicated by C.R. Doering

Keywords:
Recurrence quantification analysis
Recurrence plot
Determinism
Approximation
Phase space discretization

ABSTRACT

Given a trajectory of length N, recurrence quantification analysis (RQA) traditionally operates on the recurrence plot, whose calculation requires quadratic time and space ($\mathcal{O}(N^2)$), leading to expensive computations and high memory usage for large N. However, if the similarity threshold ε is zero, we show that the recurrence rate (*RR*), the determinism (*DET*) and other diagonal line based RQA-measures can be obtained algorithmically taking $\mathcal{O}(N\log(N))$ time and $\mathcal{O}(N)$ space. Furthermore, for the case of $\varepsilon > 0$ we propose approximations to the RQA-measures that are computable with same complexity. Simulations with autoregressive systems, the logistic map and a Lorenz attractor suggest that the approximation error is small if the dimension of the trajectory and the minimum diagonal line length are small. When applying the approximate determinism to the problem of detecting dynamical transitions we observe that it performs as well as the exact determinism measure.

1. Introduction

Recurrence quantification analysis (RQA), i.e., the quantification of structures in recurrence plots, has established in several fields of research as a powerful tool to investigate recurrence related properties of complex dynamical systems [2]. The popularity of RQA is founded on its simplicity and flexibility to be applied to almost any type of data, including non-stationary processes [3]. In particular, the outstanding role of the RQA-measure *determinism* (DET) has been demonstrated in several applications, including discriminating signals from noise [4], detecting dynamical transitions [5,6], and the recently proposed use for pattern mining and classification [7]. A comprehensive overview of recurrence plots and its applications is given in [1].

The computation and quantification of recurrence plots generally involves operations with quadratic time and space complexity ($\mathcal{O}(N^2)$). This computational complexity leads to strongly increasing computation times and memory consumption for long time series (longer than 100,000 data points). Recurrence analysis of long time series, such as audio data [8], epileptic seizures [9], material damage detection [10], or hourly weather variability [11], is, therefore, limited. Another application that can be limited by the high computational complexity is online monitoring of data streams, e.g., for video surveillance [12], monitoring social interactions [13], or assessing driving behavior [7]. This is also crucial for medical applications. For example, monitoring the brain activity of epilepsy patients by multichannel electroencephalography (EEG) can help to identify early signs of a coming epileptic seizure. This provides the opportunity to initiate an EEG-triggered on-demand countermeasure just before the epileptic seizure (e.g., by an electrical stimulation in order to reset the brain dynamics) and, hence, to improve the life quality of such patients [14,15]. Recurrence plot based measures are promising candidates for such purpose [16–19]. Similar efforts can also be found for early detection of life-threating cardio-vascular diseases, such as ventricular tachycardia [20] or obstructive sleep apnea [21]. Parallel computing approaches (e.g., using GPU calculations [11,22]) can accelerate computation but do not reduce the computational cost.

In this Letter we show the following. If the similarity threshold ε is zero, then the recurrence rate, the determinism and other diagonal line based RQA-measures are in the computational complexity class $\mathcal{O}(N\log(N))$, whereas space complexity is $\mathcal{O}(N)$. We use this

* Corresponding author.
E-mail address: schultz@dai-lab.de (D. Schultz).

http://dx.doi.org/10.1016/j.physleta.2015.01.033
0375-9601/

observation in order to propose approximations to these measures for the case of $\varepsilon > 0$. The (approximative) measures are obtained algorithmically, without having to calculate the recurrence plot.

2. Motivation

Recent work has introduced recurrence plot-based distance measures, which can be utilized for mining (multi-dimensional) time series with nonlinear dynamics [23,24]. However, the quadratic time and space complexity of computation and quantification of recurrence plots makes distance calculations for relatively long time series and online processing of fast time series streams intractable. For these purposes we aim to approximate the proposed recurrence plot-based distance measures in such a way as to reduce the computational complexity while maintaining the classification accuracy.

3. Recurrence quantification analysis

For a given d-dimensional phase space trajectory $\vec{x}$ (reconstructed from a time series x, e.g., by time-delay embedding [25]) of length N and similarity threshold $\varepsilon \geq 0$ the recurrence plot of $\vec{x}$ is an illustration of the binary recurrence matrix $\mathbf{R}$, given by

$$R_{i,j} = \Theta\big(\varepsilon - \|\vec{x}_i - \vec{x}_j\|\big), \quad i, j = 1, \ldots, N,$$

where $\|\cdot\|$ is a norm in the phase space of $\vec{x}$ and Θ is the Heaviside step function, defined by $\Theta(y) = 1$ if $y \geq 0$ and $\Theta(y) = 0$ if $y < 0$. Thus Θ indicates whether $\vec{x}_i$ and $\vec{x}_j$ are in ε-proximity (also denoted as similar) or not, i.e., $R_{i,j} = 1$ if $\|\vec{x}_i - \vec{x}_j\| \leq \varepsilon$ and $R_{i,j} = 0$ if $\|\vec{x}_i - \vec{x}_j\| > \varepsilon$. This relation is essential for the study of recurrence plots and will be used extensively in this Letter. The recurrence plot contains the *line of identity (LOI)*, which means that each entry on the main diagonal of $\mathbf{R}$ is 1. Structures parallel to the main diagonal, referred to as diagonal lines, are caused by similarly evolving epochs of the phase space trajectory $\vec{x}$.

Recurrence quantification analysis was developed in order to quantitatively describe recurrence plots. For this purpose, small scale structures, such as recurrence points or diagonal lines in the recurrence plot are used [26]. The fraction of recurrence points in the recurrence plot is measured by the *recurrence rate*,

$$RR = \frac{1}{N^2} \sum_{i,j=1}^{N} R_{i,j}, \tag{1}$$

which is interpreted as the probability to find a recurrence of trajectory $\vec{x}$. A more sophisticated RQA-measure is the *determinism*, which is defined for a given minimum diagonal line length μ as

$$DET^{(\mu)} = \frac{\sum_{l=\mu}^{N} l \cdot P(l)}{\sum_{i,j=1}^{N} R_{i,j}}, \tag{2}$$

where $P(l)$ is the number of diagonal lines of length l in $\mathbf{R}$. *DET* can be interpreted as the probability that a recurrence point belongs to a diagonal line. The parameter μ is usually set to 2. This choice is sufficient for most applications. However, in particular cases, larger values of μ can be necessary, e.g., reducing effects of tangential motion (oversampling), noise, or embedding effects [1].

As already mentioned, a phase space trajectory of a univariate time series can be reconstructed by time delay embedding [25]. We call this procedure *time series embedding*, since it is applied to the time series. In the sequel we will apply the method of time delay embedding to the trajectory $\vec{x}$ (that possibly was created by time series embedding for reconstruction purposes), but with the intention of quantifying diagonal structures in $\mathbf{R}$. In order to distinguish that from the time series embedding, we will denote this as *trajectory embedding*. More precisely, for a fixed time delay 1 and embedding dimension ν, we consider the trajectory embedding vectors

$$\vec{x}_j^{\,\nu} = (\vec{x}_j, \vec{x}_{j+1}, \ldots, \vec{x}_{j+\nu-1}), \tag{3}$$

which are of dimension $d \cdot \nu$, provided that the trajectory $\vec{x}$ is d-dimensional. The trajectory embedding of $\vec{x}$ is then defined to be the sequence $\vec{x}^{\,\nu} = (\vec{x}_j^{\,\nu})_{j=1,\ldots,N-\nu+1}$, which can be imagined as a trajectory in a $(d \cdot \nu)$-dimensional phase space. In Section 4.2 we show that information about $P(l)$ can be extracted by these representations leading to a surprising identity for the determinism.

4. *RR* and *DET* identities

We deduce identities for *RR* and $DET^{(\mu)}$, which allow fast calculation (without computing the recurrence plot) if the similarity threshold ε is zero. The identity for *RR* does hold for $\varepsilon = 0$ only. The identity for $DET^{(\mu)}$ is first shown for arbitrary $\varepsilon \geq 0$ and the assumption that the phase space norm is the maximum norm $\|\cdot\|_\infty$. However, in the special case of $\varepsilon = 0$, we will argue that the restriction to the $\|\cdot\|_\infty$-norm becomes redundant. Consequently it follows the important fact that the recurrence rate and the determinism are in $\mathcal{O}(N \log(N))$ if $\varepsilon = 0$, whereas the computational complexity of the classical methods that quantify the recurrence plot is $\mathcal{O}(N^2)$.

4.1. Recurrence rate identity

Given the trajectory embedding $\vec{x}^{\,\nu}$, Eq. (3), in analogy to Eq. (1) we define

$$\mathcal{PP}^{(\nu)} := \sum_{i,j=1}^{N-\nu+1} \Theta\big(\varepsilon - \big\|\vec{x}_i^{\,\nu} - \vec{x}_j^{\,\nu}\big\|\big), \tag{4}$$

the number of pairwise proximities of the elements in $\vec{x}^{\nu}$. Note that $RR = \mathcal{PP}^{(1)}/N^2$ is the recurrence rate of $\vec{x}$ and more general $\mathcal{PP}^{(\nu)}/(N-\nu+1)^2$ is the recurrence rate of $\vec{x}^{\nu}$.

If nominal recurrences [27] are in demand, that is $\varepsilon = 0$, then $\mathcal{PP}^{(\nu)}$ (and thus the recurrence rate RR) can be determined efficiently, i.e., with algorithmic complexity of $\mathcal{O}(N\log(N))$. In order to achieve this complexity, we employ the histogram h_X of the trajectory embedding vectors $X := \vec{x}^{\nu}$, which is given by

$$h_X : Y \to \mathbb{N}, \quad \vec{y} \mapsto \sum_{\vec{x}\in X} \Theta\big(-\|\vec{x}-\vec{y}\|\big),$$

where Y is the set of unique members of X.

Theorem 1. *Let $X = \vec{x}^{\nu}$ be the sequence of trajectory embedding vectors as defined in Eq. (3) and denote by h_X the histogram of the elements in X. If $\varepsilon = 0$, then*

$$\mathcal{PP}^{(\nu)} = \sum_{\vec{y}\in Y} \big(h_X(\vec{y})\big)^2. \tag{5}$$

Proof. First note that a similarity (or proximity) corresponds to an equality if $\varepsilon = 0$, that is

$$\Theta\big(-\|\vec{x}_i^{\nu} - \vec{x}_j^{\nu}\|\big) = 1 \quad \Leftrightarrow \quad \vec{x}_i^{\nu} = \vec{x}_j^{\nu}.$$

The claim follows by simple combinatorial arguments. Assume that for $\vec{y} \in Y$ there are exactly n elements in X that are equal to $\vec{y}$. Then there are n^2 pairwise equalities of these n elements, and hence n^2 pairwise proximities that increase $\mathcal{PP}^{(\nu)}$ by n^2. But n is exactly determined by $h_X(\vec{y}) = n$. Taking the sum over all $\vec{y} \in Y$ yields the claim. □

Based on this observation we can calculate the right-hand side of Eq. (5) efficiently. The algorithmic details are discussed in Section 5.2.2.

4.2. Determinism identity

For the rest of this Letter we choose the phase space norm $\|\cdot\|_\infty$, in particular, we assume that $\mathbf{R}$ and all $\mathcal{PP}^{(\nu)}$ are obtained for $\|\cdot\| = \|\cdot\|_\infty$. Then there is a relation between diagonal lines in the recurrence plot and recurrence points of trajectory embeddings. Before we formulate the determinism identity, we will give an intuition for the just mentioned relation: For a trajectory $\vec{x}$ let $\mathbf{R}$ be the recurrence plot. Consider the trajectory embedding $\vec{x}^2$ of $\vec{x}$ of dimension $\nu = 2$ and the corresponding recurrence plot $\mathbf{R}^{(2)}$. Now, in the maximum norm, we have that $R^{(2)}_{i,j} = 1$ is equivalent to $R_{i,j} = R_{i+1,j+1} = 1$. In other words, a diagonal line of length 2 in $\mathbf{R}$ corresponds to a recurrence point in $\mathbf{R}^{(2)}$, which is quantified by $\mathcal{PP}^{(2)}$.

Theorem 2. *Let μ be a choice of the minimum diagonal line length. For a trajectory $\vec{x}$, let the recurrence plot $\mathbf{R}$ and the pairwise proximity measures $\mathcal{PP}^{(1)}$, $\mathcal{PP}^{(\mu)}$, $\mathcal{PP}^{(\mu+1)}$ be obtained for $\|\cdot\| = \|\cdot\|_\infty$. Then for arbitrary $\varepsilon \geq 0$ it holds*

$$DET^{(\mu)} = \frac{\mu\cdot\mathcal{PP}^{(\mu)} - (\mu-1)\cdot\mathcal{PP}^{(\mu+1)}}{\mathcal{PP}^{(1)}}. \tag{6}$$

Proof. See Appendix A. □

In some cases the *LOI* of the recurrence plot should not be included in the histogram $P(l)$, i.e., $P(N)$ is set to zero. Then Theorem 2 holds true with a slight modification:

$$DET^{(\mu)} = \frac{\mu\cdot\mathcal{PP}^{(\mu)} - (\mu-1)\cdot\mathcal{PP}^{(\mu+1)} - N}{\mathcal{PP}^{(1)}}.$$

For further considerations we assume that the *LOI* is included.

It is important to discuss the condition on the underlying phase space norm that compares the elements in $\vec{x}$. First of all, the statement from Theorem 2 only holds for the $\|\cdot\|_\infty$-norm. Depending on the application, a specific norm may be selected. Usually, the Euclidean norm $\|\cdot\|_2$ is considered, but also the maximum norm $\|\cdot\|_\infty$ is often used because it is computationally faster and allows to study recurrence plots analytically [1]. If $\varepsilon = 0$, then the statement holds for all norms since each norm $\|\cdot\|$ only indicates if $\vec{x}_i$ and $\vec{x}_j$ are equal, i.e., by definition of a norm we have that $\Theta(-\|\vec{x}_i - \vec{x}_j\|) = 1$ is equivalent to $\vec{x}_i = \vec{x}_j$.

Two observations from the proof of Theorem 2 describing the relation between $P(l)$ and $\mathcal{PP}^{(\mu)}$ are worth mentioning here. Firstly,

$$\sum_{l\geq\mu} P(l) = \mathcal{PP}^{(\mu)} - \mathcal{PP}^{(\mu+1)}, \tag{7}$$

which is the number of diagonal lines in $\mathbf{R}$ of minimal length μ, and secondly

$$\mathcal{PP}^{(\nu)} = \sum_{l\geq\nu} (l-\nu+1)P(l).$$

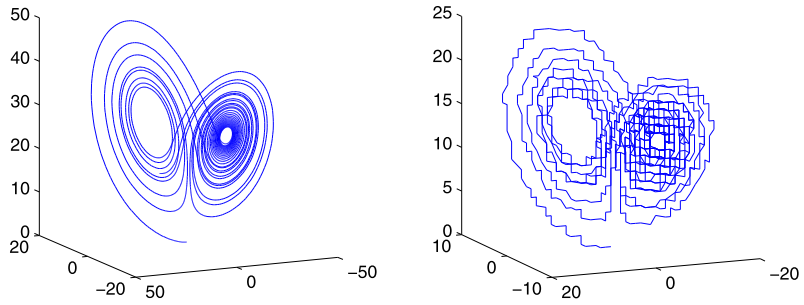


Fig. 1. The Lorenz attractor (left) from Eq. (12) and its discretization (right) for grid size parameter $\delta = 2$.

By now, the identity in Theorem 2 does not provide a method to compute the determinism efficiently for general ε. However, if $\varepsilon = 0$, then $\mathcal{PP}^{(1)}, \mathcal{PP}^{(\mu)}$ and $\mathcal{PP}^{(\mu+1)}$ can be calculated fast, as argued in Section 4.1, and then $DET^{(\mu)}$ is a simple algebraic computation in terms of these quantities.

It is worth to mention that the relationship between the length of diagonal lines in the recurrence plot and the embedding dimension is of more fundamental nature. For example, the K_2 entropy can be directly estimated from the recurrence plot using the diagonal line lengths [1] instead of the dimension of the embedding dimension [28].

5. Approximation of RQA

Approximations for RR and $DET^{(\mu)}$ are presented that are computable in $\mathcal{O}(N\log(N))$. These approximative measures are obtained algorithmically, that means we do not calculate the recurrence plot. In Section 4, we have discussed the simplified case of $\varepsilon = 0$, where these measures are in the just mentioned complexity class. In this section we study the case of $\varepsilon > 0$, for which we propose a phase space discretization approach in order to approximate $\mathcal{PP}^{(\nu)}$. The discretization will generate the situation of a zero threshold, which allows us to apply the results from Section 4.

5.1. Approximation method

We propose to discretize the phase space for a grid size parameter $\delta > 0$ via

$$\Phi_\delta : \mathbb{R}^n \to \mathbb{Z}^n, \quad \vec{y} \mapsto \tilde{\vec{y}} := \left\lfloor \frac{\vec{y}}{\delta} \right\rfloor, \tag{8}$$

where n is an arbitrary natural number and $\lfloor\cdot\rfloor$ is the component-wise round off operation. Applying Φ_δ to the trajectory $\vec{x}$ leads to a partition of the phase space in hypercubes of size δ. Then we replace the similarity condition $\|\vec{x}_i^\nu - \vec{x}_j^\nu\|_\infty \le \varepsilon$ by affiliation to the same cube, i.e., by the condition $\tilde{\vec{x}}_i^\nu = \tilde{\vec{x}}_j^\nu$. For convenience, we formulate this as a classification problem following the rules: $\vec{x}_i^\nu$ and $\vec{x}_j^\nu$ are classified as

(1) Similar if $\Theta(-\|\tilde{\vec{x}}_i^\nu - \tilde{\vec{x}}_j^\nu\|_\infty) = 1$.
(2) Dissimilar if $\Theta(-\|\tilde{\vec{x}}_i^\nu - \tilde{\vec{x}}_j^\nu\|_\infty) = 0$.

This point of view leads to the idea of proposing an approximation of $\mathcal{PP}^{(\nu)}$ for $\varepsilon > 0$ by replacing $\Theta(\varepsilon - \|\vec{x}_i^\nu - \vec{x}_j^\nu\|_\infty)$ by $\Theta(-\|\tilde{\vec{x}}_i^\nu - \tilde{\vec{x}}_j^\nu\|_\infty)$ in Eq. (4):

Definition 1. Let $\varepsilon > 0$. The approximations $\widetilde{\mathcal{PP}}^{(\nu)}$ and $\widetilde{DET}^{(\mu)}$ of $\mathcal{PP}^{(\nu)}$ and $DET^{(\mu)}$ respectively are defined as

$$\widetilde{\mathcal{PP}}^{(\nu)} := \sum_{i,j=1}^{N-\nu+1} \Theta\left(-\left\|\tilde{\vec{x}}_i^\nu - \tilde{\vec{x}}_j^\nu\right\|_\infty\right),$$

$$\widetilde{DET}^{(\mu)} := \frac{\mu \cdot \widetilde{\mathcal{PP}}^{(\mu)} - (\mu-1)\cdot \widetilde{\mathcal{PP}}^{(\mu+1)}}{\widetilde{\mathcal{PP}}^{(1)}}.$$

The crucial difference between $\mathcal{PP}^{(\nu)}$ and $\widetilde{\mathcal{PP}}^{(\nu)}$ is that for the latter the similarity threshold is zero. In this case $\widetilde{\mathcal{PP}}^{(\nu)}$ can be calculated algorithmically by applying Theorem 1 for $X = \tilde{\vec{x}}^\nu$ (rather than $X = \vec{x}^\nu$). Then $\widetilde{DET}^{(\mu)}$ simply utilizes $\widetilde{\mathcal{PP}}^{(\nu)}$ for $\nu = 1, \mu, \mu+1$ in Theorem 2.

At this point we emphasize that the approximation method and resulting approximation errors are based on the discretization only. Once we have discretized the data and use a threshold that is zero, the results from Section 4 are applied in order to calculate the RQA measures efficiently. Quantifying the discretized data with the use of a recurrence plot will lead to the exact same result. An example of a discretization is illustrated in Fig. 1. The reader is invited to follow the example in Appendix B parallel to further investigations.

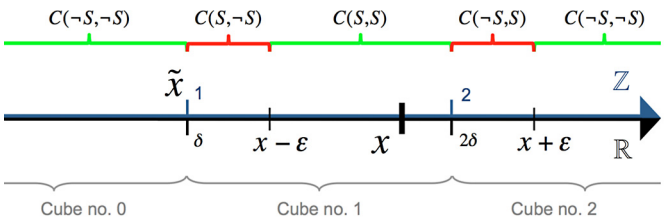


Fig. 2. Classification situations for $x \in [1.5\delta, 2\delta)$ and $\delta = 2\varepsilon$. In this one-dimensional case, the hypercubes are simply intervals in $\mathbb{R}$. Here, $\tilde{x} = 1$ and thus x belongs to the cube no. 1. For $y \in \mathbb{R}$, in fact x and y are similar if $y \in [x-\varepsilon, x+\varepsilon]$, hence x and y are not classified correctly if $y \in [\delta, x-\varepsilon)$ or $y \in [2\delta, x+\varepsilon]$.

5.2. Investigation of the approximation method

We explore the phase space discretization from Section 5.1 and its impact on the approximation of $\mathcal{PP}^{(\nu)}$. Recall that we formulated the approximation procedure as a classification problem.

Denote by $(x, y) \sim C(S, T)$ the situation that x and y are classified as belonging to class S where they are in fact in class T. Then, if S means 'similar', there are four classification situations (compare with Fig. 2), namely

$$
\begin{aligned}
(x, y) \sim C(S, S) \quad &\Leftrightarrow \quad \tilde{x} = \tilde{y} \quad \text{and} \quad \|x - y\|_\infty \leq \varepsilon. \\
(x, y) \sim C(\neg S, \neg S) \quad &\Leftrightarrow \quad \tilde{x} \neq \tilde{y} \quad \text{and} \quad \|x - y\|_\infty > \varepsilon. \\
(x, y) \sim C(S, \neg S) \quad &\Leftrightarrow \quad \tilde{x} = \tilde{y} \quad \text{and} \quad \|x - y\|_\infty > \varepsilon. \\
(x, y) \sim C(\neg S, S) \quad &\Leftrightarrow \quad \tilde{x} \neq \tilde{y} \quad \text{and} \quad \|x - y\|_\infty \leq \varepsilon.
\end{aligned}
$$

For $\vec{x}^\nu$ we conclude the following observations.

1. If for each pair $(\vec{x}_i^\nu, \vec{x}_j^\nu) \sim C(S, S)$ or $(\vec{x}_i^\nu, \vec{x}_j^\nu) \sim C(\neg S, \neg S)$, then clearly $\widetilde{\mathcal{PP}}^{(\nu)} = \mathcal{PP}^{(\nu)}$. However,
2. if $(\vec{x}_i^\nu, \vec{x}_j^\nu) \sim C(\neg S, S)$, the similarity of $(\vec{x}_i^\nu, \vec{x}_j^\nu)$ increases $\mathcal{PP}^{(\nu)}$, but not $\widetilde{\mathcal{PP}}^{(\nu)}$; and
3. if $(\vec{x}_i^\nu, \vec{x}_j^\nu) \sim C(S, \neg S)$, the dissimilarity of $(\vec{x}_i^\nu, \vec{x}_j^\nu)$ increases $\widetilde{\mathcal{PP}}^{(\nu)}$, but not $\mathcal{PP}^{(\nu)}$.

Therefore these two types of errors satisfy a mutual canceling property, and if the number of $C(\neg S, S)$-errors equals the number of $C(S, \neg S)$-errors, then even $\widetilde{\mathcal{PP}}^{(\nu)} = \mathcal{PP}^{(\nu)}$ follows.

From these considerations we establish the choice of $\delta = 2\varepsilon$.

5.2.1. The discretization parameter δ

The grid size δ of the discretization determines which elements are classified as similar and thus has to be chosen carefully. If we make no further assumptions to the data, by intuition $\delta = 2\varepsilon$ is a reasonable choice since the similarity diameter in phase space is 2ε, and moreover the different error zones have exactly the same measure (see Fig. 2). This also means that $\delta = 2\varepsilon$ is optimal and leads to nearly zero approximation error if the values of the time series x are independent uniformly distributed (on an appropriate interval). Note that $\varepsilon > 0$ was supposed implicitly since $\delta > 0$ is required in Eq. (8). If $\varepsilon = 0$, then no discretization is applied, and in fact not necessary since from Theorem 1 follows that the exact quantity $\mathcal{PP}^{(\nu)}$ can be calculated efficiently. Let us now discuss the algorithmic details.

5.2.2. Algorithms

The previous findings are used to provide algorithms for the calculation of the approximations from Definition 1; and in case of $\varepsilon = 0$ for fast calculation of the exact measures $\mathcal{PP}^{(\nu)}$ and $DET^{(\mu)}$. Since the methods for fast processing of the approximations and the exact terms are identical, for $\varepsilon = 0$ we now denote $\tilde{\vec{x}}^\nu := \vec{x}^\nu$ and state algorithms for $\widetilde{\mathcal{PP}}^{(\nu)}$ and $\widetilde{DET}^{(\mu)}$, given an arbitrary $\varepsilon \geq 0$.

As already observed in Section 4.1 it is enough to find the histogram h_X of the (discretized) sequence of trajectory embedding vectors $X := \tilde{\vec{x}}^\nu$, since then $\widetilde{\mathcal{PP}}^{(\nu)}$ is given by

$$
\widetilde{\mathcal{PP}}^{(\nu)} = \sum_{\vec{y} \in Y} \big(h_X(\vec{y})\big)^2, \tag{9}
$$

where Y is again the set of unique members of X. Technically, this may be achieved by assigning unique identifiers to the elements in X, i.e., we are interested in integers $J_1, \ldots, J_{N-\nu+1}$, such that

$$
\tilde{\vec{x}}_i^\nu = \tilde{\vec{x}}_j^\nu \quad \Leftrightarrow \quad J_i = J_j \quad \text{for all } i, j,
$$

and charge the histogram of these identifiers (compare with Algorithm 1). The calculation of $\widetilde{DET}^{(\mu)}$ is presented in Algorithm 2. Finally, the efficiency of these procedures is argued in Section 5.2.3.

Recall the designations. For more clarity, we eliminate the vector arrows in the algorithms, i.e., $x := \vec{x}$ is the trajectory of length N, $\varepsilon \geq 0$ is the similarity threshold, μ the minimum diagonal line length, ν is the trajectory embedding dimension and $x^\nu := \vec{x}^\nu$ is the matrix that consists of the rows $x_j^\nu := \vec{x}_j^\nu$, $j = 1, \ldots, N - \nu + 1$. We emphasize that the algorithm is not restricted to one-dimensional trajectories x, provided appropriate implementation. In Section 6 we provide MATLAB® code that handles multi-dimensional data.

Algorithm 1 Fast calculation of $\widetilde{\mathcal{PP}}^{(\nu)}$ (or $\mathcal{PP}^{(\nu)}$ if $\varepsilon = 0$).

1: **procedure** PPapprox(x, ε, ν)
2: **if** $\varepsilon = 0$ **then** ▷ No discretization, method is exact.
3: $\tilde{x} \leftarrow x$
4: **else** ▷ Discretization of phase space, Eq. (8).
5: $\delta \leftarrow 2\varepsilon$
6: $\tilde{x} \leftarrow \Phi_\delta(x)$
7: **end if**
8: $\tilde{x}^\nu \leftarrow$ apply_trajectory_embedding($\tilde{x}, \nu$)
9: $J = (J_1, \ldots, J_{N-\nu+1}) \leftarrow$ find_unique_row_ IDs($\tilde{x}^\nu$)
10: $h \leftarrow$ histogram(J)
11: $\widetilde{\mathcal{PP}}^{(\nu)} \leftarrow \sum_i h_i^2$
12: **end procedure**

Algorithm 2 Fast calc. of $\widetilde{DET}^{(\mu)}$ (or $DET^{(\mu)}$ if $\varepsilon = 0$).

1: **procedure** DETapprox(x, ε, μ)
2: $\mathcal{PP}^{(1)} \leftarrow$ PPapprox($x, \varepsilon, 1$)
3: $\mathcal{PP}^{(\mu)} \leftarrow$ PPapprox(x, ε, μ)
4: $\mathcal{PP}^{(\mu+1)} \leftarrow$ PPapprox($x, \varepsilon, \mu + 1$)
5: $\widetilde{DET}^{(\mu)} \leftarrow (\mu \cdot \mathcal{PP}^{(\mu)} + (\mu - 1) \cdot \mathcal{PP}^{(\mu+1)})/\mathcal{PP}^{(1)}$
6: **end procedure**

5.2.3. Complexity analysis

We specify the computational complexity by means of the number of vector-level comparisons. This conforms with the specification of the complexity of a recurrence plot computation. Here $\mathcal{O}(N^2)$ indicates the number of distance calculations between pairs of vectors, regardless of their dimension. Denote by $\mathcal{O}_c$ and $\mathcal{O}_s$ the computational and space complexity respectively.

Theorem 3. *Let ν and $\mu \in \mathbb{N}$ be fixed choices of the trajectory embedding dimension and the minimum diagonal line length, respectively.*

(i) *The complexity classes of the approximations $\widetilde{\mathcal{PP}}^{(\nu)}$ and $\widetilde{DET}^{(\mu)}$ are $\mathcal{O}_c(N\log(N))$ and $\mathcal{O}_s(N)$.*
(ii) *If $\varepsilon = 0$, then the exact terms $\mathcal{PP}^{(\nu)}$ and thus the exact RQA-measures RR and $DET^{(\mu)}$ are in the complexity classes $\mathcal{O}_c(N\log(N))$ and $\mathcal{O}_s(N)$, given an arbitrary phase space norm $||\cdot||$.*

Proof. We investigate the complexity of Algorithm 1. The complexity class of Algorithm 2 is clearly identical.

(i) It is easy to verify that the operations in lines 2–8 are in $\mathcal{O}_c(N)$ and $\mathcal{O}_s(N)$. The main cost is taken by line 9. One way to find unique identifiers for the rows of $\tilde{\tilde{x}}^\nu$ is based on sorting the rows lexicographically. Recall that the rows are trajectory embedding vectors of dimension $d \cdot \nu$. Provided a fast comparison-based sorting algorithm, e.g., Quicksort, we need $\mathcal{O}_c(N\log(N))$ vector-level comparisons of the trajectory embedding vectors in order to sort them. For completeness, each comparison of a pair of vectors needs at most $d \cdot \nu$ scalar-level comparisons in order to determine which vector is smaller with respect to the lexicographic order. However, as mentioned above, we specify the complexity at vector-level. Also note that there are sorting algorithms at scalar-level that are faster than $\mathcal{O}_c(d\nu N\log(N))$ [29]. Once the rows are sorted, it is enough to incrementally iterate the rows in order to assign an ID J_i to each row $\tilde{\tilde{x}}_i^\nu$, leading to a complexity of $\mathcal{O}_c(N)$ for the assignment step. Summarizing, the overall complexity in line 9 is $\mathcal{O}_c(N\log(N))$. In line 10 it is enough to incrementally count equal entries in J, giving $\mathcal{O}_c(N)$. Finally the complexity in line 11 is $\mathcal{O}_c(N)$ since $n \leq N$, where n is the length of the vector h. Altogether the dominating complexity classes are $\mathcal{O}_c(N\log(N))$ and $\mathcal{O}_s(N)$.

(ii) Let $\varepsilon = 0$. Determine $\mathcal{PP}^{(1)}$ using Algorithm 1 and set $RR = \mathcal{PP}^{(1)}/N^2$. Compute $DET^{(\mu)}$ using Algorithm 2. By Theorems 1 and 2 these expressions coincide with the exact RQA-measures. As already mentioned in Section 4.2, if $\varepsilon = 0$, then the identities hold for an arbitrary phase space norm since each norm only indicates whether two elements in phase space are equal. The claim on the complexity classes is proven in the first part. □

We remark that sorting the rows lexicographically is not the only possibility. One could, for instance, use a hash function that maps the embedding vectors to $\mathbb{R}$ in order to get the identifiers for the embedding vectors and then apply a simple one-dimensional sorting algorithm to find the histogram incrementally. However, such hash functions do not guarantee unique identifiers since they are not injective in general.

5.2.4. Worst case error

As shown in Theorem 3, if $\varepsilon = 0$, then $\mathcal{PP}^{(\nu)}$ can be calculated exactly and efficiently. If $\varepsilon > 0$, the approximation $\widetilde{\mathcal{PP}}^{(\nu)}$ of $\mathcal{PP}^{(\nu)}$ satisfies the following estimates.

Theorem 4. *Let $\varepsilon > 0$ and $\delta = 2\varepsilon$. In d-dimensional phase space it holds*

$$\frac{1}{2^{d\nu}}\mathcal{PP}^{(\nu)} \leq \widetilde{\mathcal{PP}}^{(\nu)} \leq 2^{d\nu}\mathcal{PP}^{(\nu)}.$$

Proof. Denote $m = d\nu$. The lower bound is reached if the number of $C(\neg S, S)$-errors is maximal. Let $\vec{y}$ be a vertex of the discretization lattice. In m-dimensional space there are 2^m adjoint hypercubes surrounding $\vec{y}$. Hence it is possible to place 2^m points $\vec{x}_i$, each in another cube, such that $\|\vec{y} - \vec{x}_i\|_\infty \leq \varepsilon/2$ for all i. It follows that $\|\vec{x}_i - \vec{x}_j\|_\infty \leq \varepsilon$ for all i, j. Hence each pair $(\vec{x}_i, \vec{x}_j)$ is similar, but by construction

classified as dissimilar if $i \neq j$. In this case we have $\mathcal{PP}^{(\nu)} = (2^m)^2$ and $\widetilde{\mathcal{PP}}^{(\nu)} = 2^m$. The argument is finished since placing additional points only leads to a reduction of the number of $C(\neg S, S)$-errors.

The upper bound follows in a similar manner by producing errors of type $C(S, \neg S)$. □

By now the bounds are shown to be existent (hence the theorem is true) but not that they are sharp. One would have to show that there is a trajectory whose embedding vectors are constructed as above. For $\nu = 2$ and $d = 1$ an appropriate trajectory is given by $\vec{x} = (\eta, \eta, -\eta, -\eta, \eta)$, where $\eta < \varepsilon/2$. The four resulting trajectory embedding vectors of $\vec{x}$ satisfy the above construction. For general ν and d this becomes more technical, but we think that this investigation is unnecessary at this point. It is more interesting how the approximation error behaves empirically.

5.2.5. Empirical approximation error

As seen in Section 5.2.4 the bounds of the approximation error of $\widetilde{\mathcal{PP}}^{(\nu)}$ are rather large and monotonic in ν. However, the constructions given in the proof of Theorem 4 to reach these bounds are very specific.

In this section we study the approximation errors of $\widehat{\mathcal{PP}}^{(\nu)}$ and $\widetilde{DET}^{(\mu)}$ empirically. For this sake the relative mean errors of 100 realizations, designed as follows, are determined. For each experiment the autoregressive process $\vec{x} = (x_1, \ldots, x_N)$, with

$$x_i = ax_{i-1} + b\eta_i, \quad i = 2, \ldots, N, \tag{10}$$

is generated for $N = 1000$ time steps, where $x_1 = 0$, a, b are fixed values that are chosen randomly independent uniformly distributed on $[0, 1]$ and η is a vector of Gaussian white noise. Then the approximations are determined by the algorithms from Section 5.2.2 and the exact quantities $\mathcal{PP}^{(\nu)}$ and $DET^{(\mu)}$ are calculated by the classical method in order to specify the accuracy of the approximations. The results are illustrated in Fig. 3 for several combinations of ν (resp. μ) and ε, where the height of the bars corresponds to the mean error and the color of the bars corresponds to the value $\mathcal{PP}^{(\nu)}$ and $DET^{(\mu)}$, respectively. It is customary to select ε as a few percent of the phase space diameter [1,30], which in this case is given by $\text{range}(\vec{x}) = \max(\vec{x}) - \min(\vec{x})$.

We observe that the approximation errors are basically increasing in ν (resp. μ) and ε. However, most of the combinations of ν (resp. μ) and ε have little relevance. First, if ε is small and ν (resp. μ) is large, the probability to find recurrences is low. Consequently the bars in Fig. 3(a) are of deep blue color. Therefore the low error in this area is an artefact. Conversely, if ν (resp. μ) is small and ε is large, too many recurrences are found, resulting in red colors. Reasonable choices of ν (resp. μ) and ε are indicated by colors in the range from blue-green to orange-red in Fig. 3(a).

As an example, assume that we want to determine the recurrence rate and the determinism of the trajectory $\vec{x}$. For the calculation of the determinism, a minimal line length of $\mu = 2$ is sufficient, because for the autoregressive process we do not expect much effect of tangential motion or sampling [1]. Then for all sensible values of ε, i.e., from 0 to 8 percent of the range, we obtain mean approximation errors below 1.4% for the recurrence rate and below 2.7% for the determinism.

It should be noticed that we have investigated one-dimensional trajectories $\vec{x}$ that are not reconstructed by time series embedding. However, the trajectory embedding of dimension ν can also be imagined as time series embedding if we postulate that $\vec{x}$ is the time series and $\vec{x}^{\nu}$ is the trajectory, which is obtained from $\vec{x}$ by time series embedding with time delay 1 and embedding dimension ν. Then Fig. 3(a) reflects the approximation errors of the recurrence rate of $\vec{x}^{\nu}$, which is given by $\mathcal{PP}^{(\nu)}/(N - \nu + 1)^2$. The essence of this technical point of view is that the approximation errors increase if the dimension of the trajectory increases. We also observe this in the experiment from Section 6 for the 3-dimensional Lorenz attractor, see Table 2.

6. Execution time of Algorithm 1

We compare the execution times of $\mathcal{PP}$ and its approximation $\widetilde{\mathcal{PP}}$ on a consumer computer (2.3 GHz Intel Core i7 quad core processor, 16 GB 1600 MHz DDR3 RAM). Since execution times do not only depend on the algorithm, but also on the implementation, we provide MATLAB® code. Note that, however, this code uses standard MATLAB® routines and may be strongly optimized by the MATLAB® compiler.

We evaluate two systems, the autoregressive process

$$x_1 = 0, \qquad x_i = 0.57x_{i-1} + 0.24\eta_i, \quad i = 2, \ldots, 100{,}000{,}000 \tag{11}$$

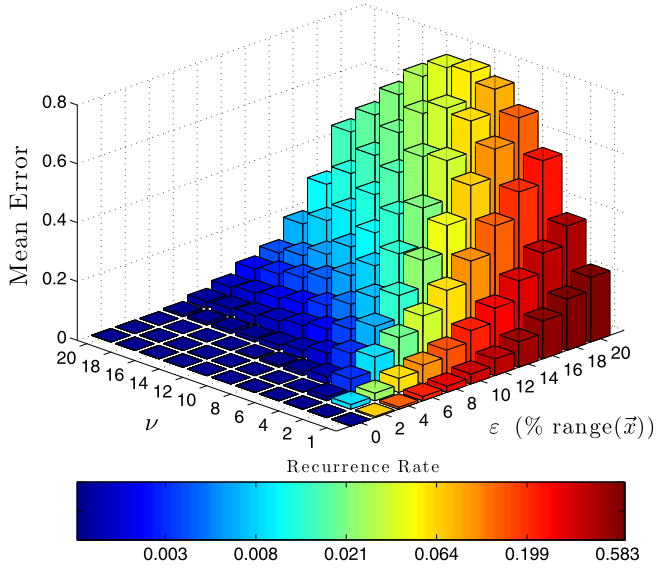
and the well-known 3-dimensional Lorenz system (see Fig. 1)

$$\dot{x} = a(y - x), \qquad \dot{y} = x(b - z) - y, \qquad \dot{z} = xy - cz \tag{12}$$

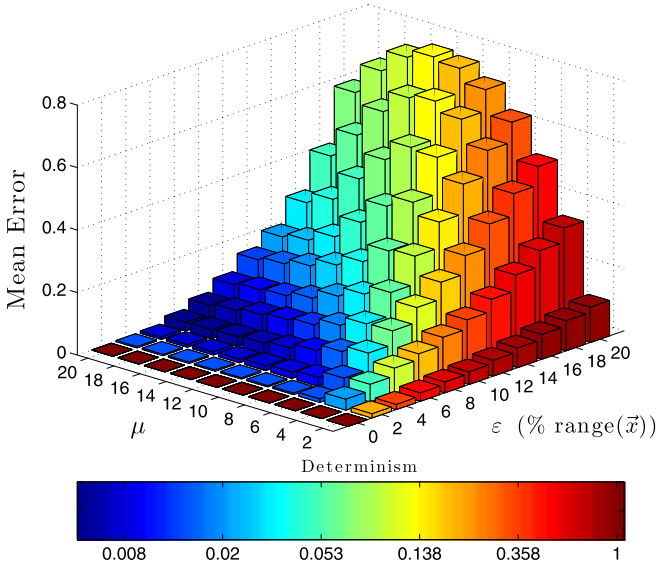
for the parameters $a = 10, b = 28, c = 8/3$. Then these systems are truncated according to the values of N as listed in the tables of results, Tables 1 and 2, and Fig. 4, and processed by the routines. The threshold ε was chosen for each N separately as 7% of the phase space diameter and no embedding is applied, i.e. $m = 1$ in the following MATLAB® function.

MATLAB® code for $\widetilde{\mathcal{PP}}$

```
function pp = PPapprox( x, eps, m)
[N,d] = size(x);
if eps > 0                              % discretize if eps > 0
    x = floor(x/(2*eps));
end
X = zeros(N-m+1,d*m);                   % apply trajectory embedding
for i = 1:m
    X(:,d*(i-1)+1:d*i) = x(i:N-(m-i),:);
```



(a) Approximation error of $\tilde{RR}^{(\nu)}$



(b) Approximation error of $\tilde{DET}^{(\mu)}$

Fig. 3. Relative mean errors obtained from 100 autoregressive process realizations. The bar color in (a) indicates the value of the exact recurrence rate $RR^{(\nu)} = \mathcal{PP}^{(\nu)}/(N-\nu+1)^2$ of the embedded trajectory $\vec{x}^{\nu}$. The bar color in (b) reflects the exact determinism $DET^{(\mu)}$ of the trajectory $\vec{x}$, given a minimum diagonal line length μ. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

```
end
[u,~,iu] = unique(X,'rows');          % find row ID's iu
h = hist(iu,size(u,1));               % find histogram of row ID's
pp = sum(h.^2);
end
```

MATLAB® code for $\mathcal{PP}$

```
function pp = PP( x, eps)
R = pdist2(x,x,'chebychev') <= eps;   % calculate recurrence plot
pp = nnz(R);                          % count nonzeros
end
```

Since the available memory on the computer was 12 GB, we limited the data size for the exact measure $\mathcal{PP}$. Indeed a single recurrence plot for $N = 40{,}000$ consumes about 12 GB of RAM, provided double precision and no storage optimization. For $N = 100{,}000$ even about 75 GB of memory would be required. The results give numerical evidence for the complexity we have proved in Theorem 3 and reflect

Table 1
Mean execution times obtained from 10 realizations of the autoregressive process (11) for different time series length N. The approximation error is again the mean over the relative errors $|\mathcal{PP} - \widetilde{\mathcal{PP}}|/\mathcal{PP}$.

N	Execution time $\mathcal{PP}$ (s)	Execution time $\widetilde{\mathcal{PP}}$ (s)	Approximation error
100	0.0552	0.0005	0.0275
1000	0.0078	0.0005	0.0104
10,000	0.9058	0.0018	0.0098
20,000	3.7314	0.0158	0.0098
30,000	8.3865	0.0233	0.0096
35,000	13.7078	0.0131	0.0092
100,000	–	0.0169	–
1,000,000	–	0.1912	–
10,000,000	–	2.2587	–
100,000,000	–	28.5899	–

Table 2
Mean execution times obtained from 10 realizations of the Lorenz system (12) for different time series length N. The approximation error is again the mean over the relative errors $|\mathcal{PP} - \widetilde{\mathcal{PP}}|/\mathcal{PP}$.

N	Execution time $\mathcal{PP}$ (s)	Execution time $\widetilde{\mathcal{PP}}$ (s)	Approximation error
100	0.0513	0.0009	0.0471
1000	0.0077	0.0006	0.3655
10,000	0.9071	0.0052	0.2885
20,000	3.7200	0.0074	0.2646
30,000	8.2962	0.0117	0.2746
100,000	–	0.0396	–
1,000,000	–	0.3645	–

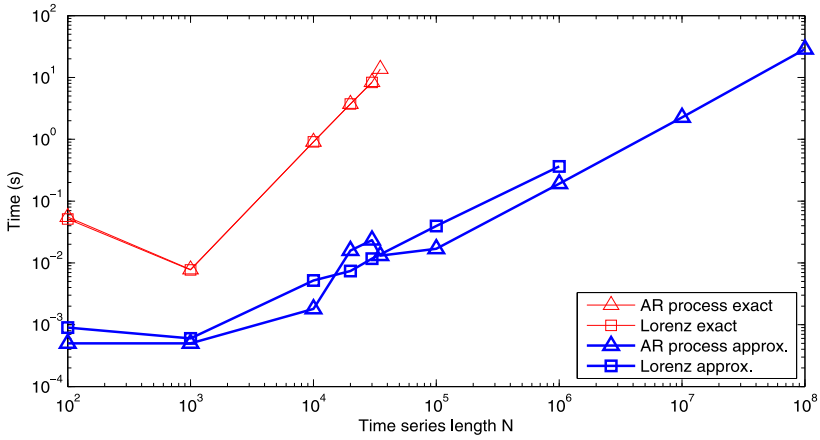


Fig. 4. Visual representation of the execution times from Tables 1 and 2 in log scale.

the large difference between $\mathcal{O}(N^2)$ and $\mathcal{O}(N\log(N))$ for increasing N. For example the ratio of execution times for the autoregressive process with $N = 35{,}000$ is about 1.046. Moreover, the algorithm is very fast for extreme large data and the approximation error decreases slightly with growing N. In Table 2 the small approximation error for $N = 100$ is due to the short and hence almost linear attractor. As expected, the other errors of the Lorenz experiment are higher since the attractor is 3-dimensional.

7. Application to transition detection

7.1. Introduction to the problem

Assume that we are given a time series or a stream $x = (x_1, x_2, x_3, \ldots)$ which changes its dynamics at unknown time segments. It has been shown that the determinism $DET^{(\mu)}$ is able to find these periods [5,6,31]. For this, the time series is analyzed window-wise for a window size w and step size s, leading to a sequence $\mathcal{D}$ of determinism-values. More precisely, $\mathcal{D}(j)$ contains the determinism of the sub-sequence $(x_{s\cdot j}, \ldots, x_{s\cdot j+w-1})$, $j = 1, 2, 3, \ldots$. A transition in the dynamics is indicated when the system leaves its typical dynamical behavior, in this case its typical range of the window-wise determinism values [6]. The bounds of this range are referred to as confidence levels. An example of a graph of $\mathcal{D}$ is illustrated as red line in Fig. 5. In the gray marked area the system from the upper plot changes its dynamics (details in Section 7.2) and consequently $\mathcal{D}$ exceeds its upper confidence bound, which is represented by the dashed red line.

In this section we compare our proposed approximation $\widetilde{DET}^{(\mu)}$ to the exact measure $DET^{(\mu)}$ for the problem of identifying transition times. Again, we consider a minimal line length of $\mu = 2$. It remains to select ε. For each window ε is determined separately such that the recurrence rate is a small fraction, e.g., 0.1 [6]. This leads to a constant (in time index i) denominator in Eq. (2) accentuating the behavior of the changes in $P(l)$.

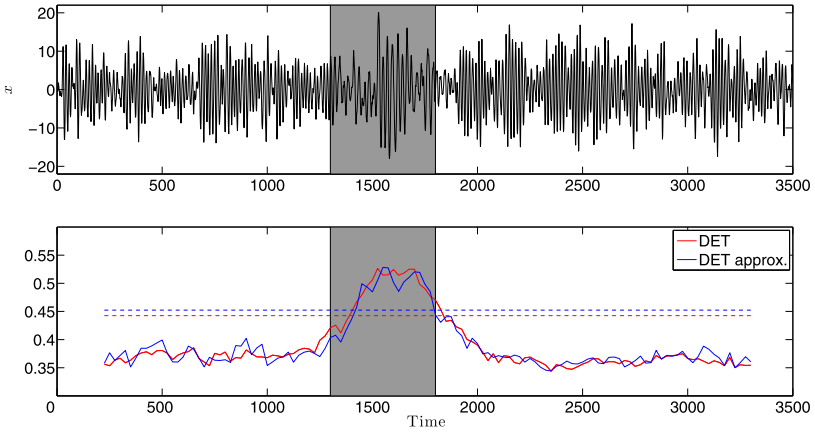


Fig. 5. Example from the transition experiment. The system of the upper plot is generated as described in Section 7.2. The lower graphic shows the window-wise determinism sequence $\mathcal{D}$ and its approximation $\widetilde{\mathcal{D}}$. The dashed lines are the confidence levels.

Table 3
Mean transition errors obtained from 100 realizations as described in Section 7.2. The left/right transition error is defined as the absolute deviation from time index 1300/1800.

Measure	Left transition error	Right transition error
$DET^{(2)}$	61.25	64.75
$\widetilde{DET}^{(2)}$	85.00	45.50

7.2. Autoregressive process with changing parameters

The experiment is inspired by [6]. We evaluate 100 realizations employing autoregressive processes of order 2,

$$x_i = ax_{i-1} - bx_{i-2} + c\eta_i.$$

The test time series is initially generated for $x_1 = x_2 = 0$ and $a = 1.8, b = 0.972, c = 0.64$ for 1300 time steps. Then the parameters change for a period of 500 time steps to $a = 1.85, b = 0.917, c = 0.76$. Finally the system returns to the initial parameters and stops at time step 3500. The resulting time series x is then analyzed for a window size $w = 400$ and step size $s = 25$.

In the exact case $\mathcal{D}$ contains the values of $DET^{(2)}$, where ε is chosen such that the recurrence rate $\mathcal{PP}^{(1)}/w^2$ is 0.1. In the approximative case $\widetilde{\mathcal{D}}$ contains the values of $\widetilde{DET}^{(2)}$, where ε is chosen such that the approximate recurrence rate $\widetilde{\mathcal{PP}}^{(1)}/w^2$ is 0.1.

In order to find the upper confidence level, we assume that the system with initial parameters is observed for $N = 100{,}000$ time steps and the distribution of the (approximate) determinism values for the windows is charged. We choose as transition level the 99.95%-quantile of those distributions, leading to a confidence level of 0.4425 for $DET^{(2)}$ and 0.4523 for $\widetilde{DET}^{(2)}$. For the evaluation the time points of exceeding and falling below these levels are compared to the actual transition time at 1300 and 1800. The results reveal that our fast approximation performs as well as the slow exact method (Table 3).

7.3. Transitions in the logistic map

We briefly illustrate that the approximate determinism $\widetilde{DET}$ is also able to find transitions in the logistic map [20,31]

$$x_{i+1} = ax_i(1 - x_i), \tag{13}$$

with control parameter a in the range $[3.6, 3.8]$ and $x_1 = 0.7$. In the simulations we observed that the quality of the approximation is sensitive to ε. We found that a rather small threshold is beneficial. More precisely, we selected ε for each a separately such that the recurrence rate resp. the approximate recurrence rate is 0.01. Fig. 6 confirms that $\widetilde{DET}$ has the capability to find dynamical transitions. For $N = 10{,}000$ and 400 iterative values of a we observed that the approximation was about 124 times as fast as the exact computation, including all iterations and determinations of ε (24.5 s for approximation, 3041.7 s for exact determinism).

8. Other RQA measures

Using $\mathcal{PP}^{(\nu)}$ it is possible to specify identities for other diagonal line based RQA-measures. Detailed analysis of those is out of the scope of this work, but we briefly state the formulas in this section. In the following equations the left-hand side is the classical definition and the right-hand side is the identity in terms of $\mathcal{PP}^{(\nu)}$. As before, μ denotes the minimum diagonal line length, the phase space norm is $\|\cdot\|_\infty$ and the *LOI* is included unless otherwise stated.

For the ratio $RATIO^{(\mu)}$ between $DET^{(\mu)}$ and *RR*, we get

$$N^2 \frac{\sum_{l\geq\mu} lP(l)}{(\sum_{l\geq 1} lP(l))^2} = N^2 \frac{\mu\mathcal{PP}^{(\mu)} - (\mu-1)\mathcal{PP}^{(\mu+1)}}{(\mathcal{PP}^{(1)})^2}.$$

Due to Eq. (7), the averaged diagonal line length $L^{(\mu)}$ is given by

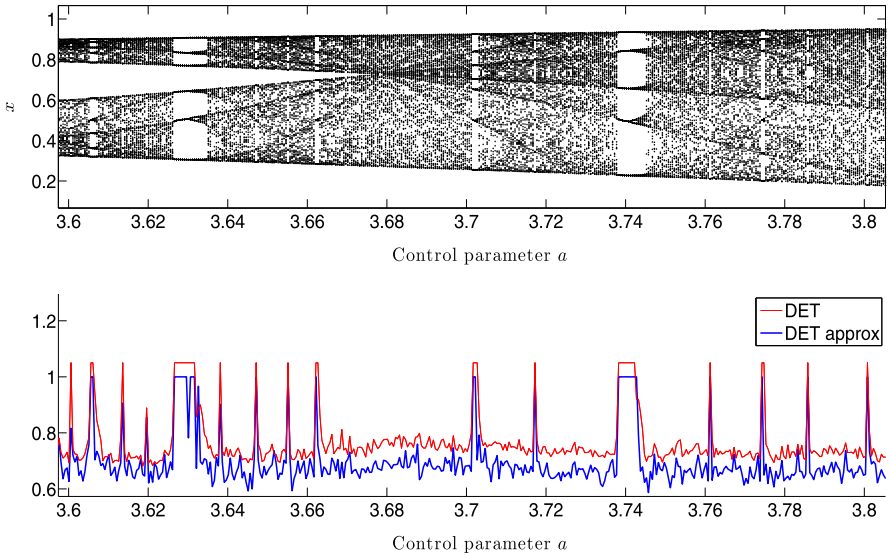


Fig. 6. Bifurcation diagram of the logistic map (13) and its dynamics. We observe multiple chaos-period transitions that are found by both measures, *DET* and its approximation $\widetilde{DET}$. For clarity, the peaks of *DET* are increased from 1 to 1.05.

$$\frac{\sum_{l\geq\mu} lP(l)}{\sum_{l\geq\mu} P(l)} = \frac{\mu\mathcal{PP}^{(\mu)} - (\mu-1)\mathcal{PP}^{(\mu+1)}}{\mathcal{PP}^{(\mu)} - \mathcal{PP}^{(\mu+1)}},$$

and the length of the longest diagonal line $L_{\max}$ (excluding the *LOI*, i.e., $P(N) := 0$), determined by

$$\max\{l \mid P(l) \neq 0\} = \min\{v \mid \mathcal{PP}^{(v)} = N - v + 1\} - 1,$$

can be found by binary search in $\mathcal{O}(\log(N))$ iterations since $v \mapsto \mathcal{PP}^{(v)}$ is monotonically decreasing.

Finally we should remember that these expressions can be calculated exactly and efficiently if $\varepsilon = 0$; and in case of $\varepsilon > 0$ the measures can be approximated by replacing $\mathcal{PP}$ by $\widetilde{\mathcal{PP}}$. In both cases Algorithm 1 determines the pairwise proximity measures efficiently. For $\widetilde{L}_{\max}$ the resulting computational complexity is $\mathcal{O}(N\log^2(N))$. All other approximative measures are in $\mathcal{O}(N\log(N))$, whereas the complexity of the classical measures is $\mathcal{O}(N^2)$ [11].

9. Conclusions

We have shown that the recurrence rate and many diagonal line based RQA-measures can be calculated efficiently, i.e., in $\mathcal{O}(N\log(N))$ if the similarity threshold ε is zero. For the case $\varepsilon > 0$ we have introduced approximations of these measures that are based on phase space discretization and a relation between the histogram of the diagonal line lengths $P(l)$ and the introduced pairwise proximity measures of trajectory embeddings. For small embedding dimension v or minimum diagonal line length μ the proposed approximations are very close to the exact quantities in our experiments with one-dimensional data, while execution times and memory consumption are significantly lower. However, we suggest to compare the approximations with the exact measures since the accuracy strongly depends on the data under study. In particular, if the trajectory is multi-dimensional the approximation error may increase. We also recommend to keep ε as small as possible since it determines the grid size of the discretization. In general, a small grid size leads to gentle discretization and, consequently, causes low approximation errors.

The proposed approximation of diagonal line based RQA-measures, with its substantially improved time and space complexity, makes it possible to efficiently analyze the nonlinear behavior of fast data streams and long data sequences, even if the computations are performed on an embedded system with limited processing power and memory. The online analysis of data streams is critical for medical applications that monitor vital functions, such as brain and heart activity. The offline processing of long data sequences is important for analyzing historical measurements, e.g., hourly weather variability, which usually cause expensive or even intractable computations if traditional RQA-measures without approximation techniques are employed.

In future work we aim at investigating data-adopted discretization lattices and their ability to improve the approximation accuracy. Furthermore, we plan to study under which conditions the ratio $\widetilde{DET}(\vec{x})/DET(\vec{x})$ stays stable (i.e. nearly constant) for changing $\vec{x}$. Clearly, if the approximation error is small, this stability is present. But if the error is large, this stability would still allow us to compare dynamics rather than determine dynamics, which is sufficient for many applications, e.g., transition detection.

Appendix A. Proof of the determinism identity

Proof of Theorem 2. Let $\vec{x} = (\vec{x}_1, \ldots, \vec{x}_N)$ be a d-dimensional phase space trajectory ($d \in \mathbb{N}$) of length N and the similarity threshold $\varepsilon \geq 0$ as well as the minimum diagonal line length $\mu \in \mathbb{N}$ be given. Assume that $\|\cdot\|_\infty$ is the underlying phase space norm, i.e., the recurrence plot of $\vec{x}$ is given by

$$R_{i,j} = \Theta\big(\varepsilon - \|\vec{x}_i - \vec{x}_j\|_\infty\big), \quad i, j = 1, \ldots, N,$$

and for all $v \in \mathbb{N}$ the pairwise proximity measures $\mathcal{PP}^{(v)}$ are defined as

$$\mathcal{PP}^{(\nu)} = \sum_{i,j=1}^{N-\nu+1} \Theta\left(\varepsilon - \left\|\vec{x}_i^{\nu} - \vec{x}_j^{\nu}\right\|_\infty\right).$$

By definition it holds $\sum_{i,j=1}^{N} R_{i,j} = \mathcal{PP}^{(1)}$, thus we have to show that the numerators in Eq. (6) are equal. We show

$$\sum_{l=\mu}^{N} l \cdot P(l) = \mathcal{PP}^{(\mu)} + (\mu - 1) \cdot \left(\mathcal{PP}^{(\mu)} - \mathcal{PP}^{(\mu+1)}\right),$$

which gives additional insights into the relation between $P(l)$ and $\mathcal{PP}$ (7). For this, we define the following index sets.

$$I^{\mu} = \left\{(i,j) \;\middle|\; \Theta\left(\varepsilon - \left\|\vec{x}_i^{\mu} - \vec{x}_j^{\mu}\right\|_\infty\right) = 1\right\}$$

$$I_l = \left\{(i,j) \;\middle|\; \sum_{k=0}^{l-1} \Theta\left(\varepsilon - \|x_{i+k} - x_{j+k}\|_\infty\right) = \sum_{k=-1}^{l} \Theta\left(\varepsilon - \|x_{i+k} - x_{j+k}\|_\infty\right) = l\right\}$$

$$J_l = \left\{(i+k, j+k) \;\middle|\; (i,j) \in I_l, k = 0, \ldots, l-1\right\}$$

$$I_l^{\mu} = I^{\mu} \cap J_l$$

I^{μ} contains exactly the index pairs (i,j) such that $\vec{x}_i^{\mu}$ and $\vec{x}_j^{\mu}$ are similar, thus $\mathcal{PP}^{(\mu)}$ is already determined by $\mathcal{PP}^{(\mu)} = \sum_{(i,j)\in I^{\mu}} \Theta(\varepsilon - \|\vec{x}_i^{\mu} - \vec{x}_j^{\mu}\|_\infty)$.

I_l is the set of index pairs (i,j) such that there is a diagonal line of exactly length l in $\mathbf{R}$ starting at (i,j), thus $P(l) = |I_l|$.

J_l is the set of index pairs (i,j) that exactly cover all diagonal lines of length l in $\mathbf{R}$, i.e., $l \cdot P(l) = |J_l|$.

Denote $\theta_k = \Theta(\varepsilon - \|x_{i+k} - x_{j+k}\|_\infty)$, then the relation between I_l^{μ} and I_l is described by

$$\begin{aligned}
(i,j) \in I_l \quad &\Leftrightarrow \quad \theta_{-1} = \theta_l = 0 \quad \text{and} \quad \Theta\left(\varepsilon - \|\vec{x}_{i+k} - \vec{x}_{j+k}\|_\infty\right) = 1 \quad \text{for all } k = 0, \ldots, l-1 \\
&\Leftrightarrow \quad \theta_{-1} = \theta_l = 0 \quad \text{and} \quad \Theta\left(\varepsilon - \left\|\vec{x}_{i+k}^{\mu} - \vec{x}_{j+k}^{\mu}\right\|_\infty\right) = 1 \quad \text{for all } k = 0, \ldots, l-\mu \\
&\Leftrightarrow \quad \theta_{-1} = \theta_l = 0 \quad \text{and} \quad \sum_{k=0}^{l-\mu} \Theta\left(\varepsilon - \left\|\vec{x}_{i+k}^{\mu} - \vec{x}_{j+k}^{\mu}\right\|_\infty\right) = l - \mu + 1 \qquad (14) \\
&\Leftrightarrow \quad (i+k, j+k) \in I_l^{\mu} \quad \text{for all } k = 0, \ldots, l-\mu. \qquad (15)
\end{aligned}$$

Moreover, note that by construction $I_{l_1}^{\mu} \cap I_{l_2}^{\mu} = \emptyset$ for $l_1 \neq l_2$ and $\bigcup_{l\geq\mu} I_l^{\mu} = I^{\mu}$:

$$I_{l_1}^{\mu} \cap I_{l_2}^{\mu} = \left(I^{\mu} \cap J_{l_1}\right) \cap \left(I^{\mu} \cap J_{l_2}\right) \subset (J_{l_1} \cap J_{l_2}) = \emptyset \quad \text{for } l_1 \neq l_2, \qquad (16)$$

$$\bigcup_{l\geq\mu} I_l^{\mu} = \bigcup_{l\geq\mu} I^{\mu} \cap J_l = I^{\mu} \cap \bigcup_{l\geq\mu} J_l = I^{\mu}. \qquad (17)$$

Now we have collected all relations to begin the actual proof. Since

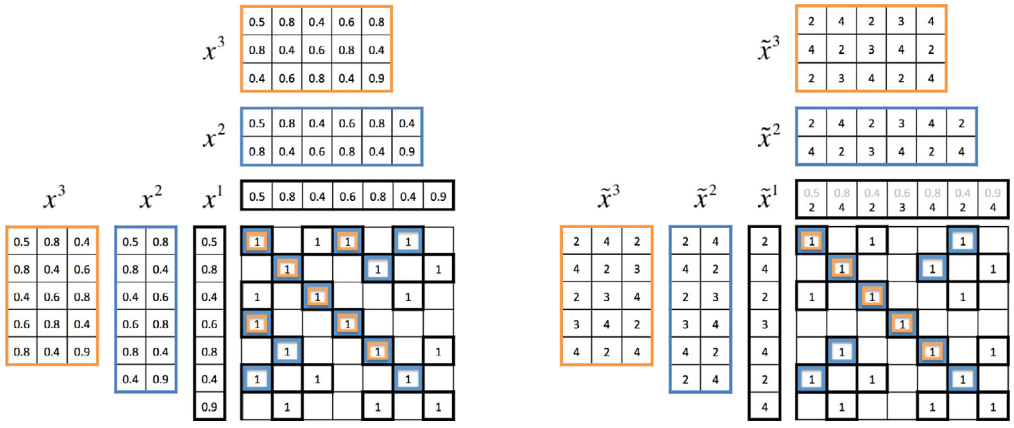
$$P(l) = |I_l| = \sum_{(i,j)\in I_l} 1 \overset{(14)}{=} \sum_{(i,j)\in I_l} \frac{\sum_{k=0}^{l-\mu} \Theta(\varepsilon - \|\vec{x}_{i+k}^{\mu} - \vec{x}_{j+k}^{\mu}\|_\infty)}{l - \mu + 1},$$

we get

$$(l - \mu + 1) P(l) = \sum_{(i,j)\in I_l} \sum_{k=0}^{l-\mu} \Theta\left(\varepsilon - \left\|\vec{x}_{i+k}^{\mu} - \vec{x}_{j+k}^{\mu}\right\|_\infty\right)$$

and therefore

$$\begin{aligned}
\sum_{l\geq\mu} (l - \mu + 1) P(l) &= \sum_{l\geq\mu} \sum_{(i,j)\in I_l} \sum_{k=0}^{l-\mu} \Theta\left(\varepsilon - \left\|\vec{x}_{i+k}^{\mu} - \vec{x}_{j+k}^{\mu}\right\|_\infty\right) \\
&\overset{(15)}{=} \sum_{l\geq\mu} \sum_{(i,j)\in I_l^{\mu}} \Theta\left(\varepsilon - \left\|\vec{x}_i^{\mu} - \vec{x}_j^{\mu}\right\|_\infty\right) \\
&\overset{(16)}{=} \sum_{(i,j)\in \bigcup_{l\geq\mu} I_l^{\mu}} \Theta\left(\varepsilon - \left\|\vec{x}_i^{\mu} - \vec{x}_j^{\mu}\right\|_\infty\right) \\
&\overset{(17)}{=} \sum_{(i,j)\in I^{\mu}} \Theta\left(\varepsilon - \left\|\vec{x}_i^{\mu} - \vec{x}_j^{\mu}\right\|_\infty\right) \\
&= \mathcal{PP}^{(\mu)}.
\end{aligned}$$



(a) $\mathbf{R}^{(\nu)}$: RPs of trajectories $\vec{x}^{\nu}$. (b) $\tilde{\mathbf{R}}^{(\nu)}$: RPs of discretized trajectories $\tilde{\vec{x}}^{\nu}$

Fig. 7. Recurrence plots (RPs) of the recurrence matrices from Eq. (19) for $\nu = 1, 2, 3$, illustrated in black, blue, orange, respectively. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Rearranging this equation leads to

$$\sum_{l\geq\mu} lP(l) = \mathcal{PP}^{(\mu)} + (\mu-1)\sum_{l\geq\mu} P(l), \tag{18}$$

and it remains to show that

$$\mathcal{PP}^{(\mu)} - \mathcal{PP}^{(\mu+1)} = \sum_{l\geq\mu} P(l).$$

But this already follows by applying (18) for μ and $\mu+1$:

$$\begin{aligned}
\mathcal{PP}^{(\mu)} - \mathcal{PP}^{(\mu+1)} &= \sum_{l\geq\mu} lP(l) - (\mu-1)\sum_{l\geq\mu} P(l) - \Bigg(\sum_{l\geq\mu+1} lP(l) - \mu \sum_{l\geq\mu+1} P(l)\Bigg) \\
&= \sum_{l\geq\mu} lP(l) - (\mu-1)\sum_{l\geq\mu} P(l) - \Bigg(\sum_{l\geq\mu} lP(l) - \mu P(\mu) - \mu\sum_{l\geq\mu} P(l) + \mu P(\mu)\Bigg) \\
&= \sum_{l\geq\mu} P(l). \qquad \square
\end{aligned}$$

Appendix B. Example

Assume that we want to compute the determinism of the sample trajectory $\vec{x} = (0.5, 0.8, 0.4, 0.6, 0.8, 0.4, 0.9)$, given a similarity threshold $\varepsilon = 0.1$. First we consider the recurrence plots

$$R^{(\nu)}_{i,j} := \Theta\big(\varepsilon - \|\vec{x}^{\nu}_i - \vec{x}^{\nu}_j\|_{\infty}\big), \qquad \widetilde{R}^{(\nu)}_{i,j} := \Theta\big(-\|\tilde{\vec{x}}^{\nu}_i - \tilde{\vec{x}}^{\nu}_j\|_{\infty}\big) \tag{19}$$

of the embedded (3) trajectory $\vec{x}^{\nu}$ and its discretization $\tilde{\vec{x}}^{\nu} = \Phi_{\delta}(\vec{x}^{\nu})$, where $\delta = 0.2$ (see (8) and Section 5.2.1). The embedded trajectories and the corresponding recurrence plots are illustrated in Fig. 7 for several embedding dimensions $\nu = 1, 2, 3$ in black, blue, and orange color, respectively. For example the recurrence plots for $\nu = 2$ comprise the recurrences marked by blue color, the black only highlighted entries are no recurrences for $\nu = 2$.

For $\nu = 1$ (write $\mathbf{R} = \mathbf{R}^{(1)}$) we observe that $R_{1,4} = 1$, but $\widetilde{R}_{1,4} = 0$. That means the pair $(0.5, 0.6)$ is similar, i.e.,

$$|0.5 - 0.6| = 0.1 \leq \varepsilon,$$

but classified as dissimilar:

$$\Phi_{\delta}(0.5) = \left\lfloor \frac{0.5}{0.2} \right\rfloor = 2 \neq 3 = \left\lfloor \frac{0.6}{0.2} \right\rfloor = \Phi_{\delta}(0.6).$$

Due to symmetry the pairs $(0.5, 0.6)$ and $(0.6, 0.5)$ lead to $C(S, \neg S)$-errors (see Section 5.2). For all other pairs the classified and actual similarity statements coincide.

Recall that the determinism is the ratio between the number of points on diagonal lines and all points in the recurrence plot. For the trajectories $\vec{x} = \vec{x}^1$ and $\tilde{\vec{x}} = \tilde{\vec{x}}^1$ we obtain by counting the black structures in the recurrence plots:

$$DET^{(2)} = \frac{17}{21} \approx 0.81, \qquad \widetilde{DET}^{(2)} = \frac{15}{19} \approx 0.79. \tag{20}$$

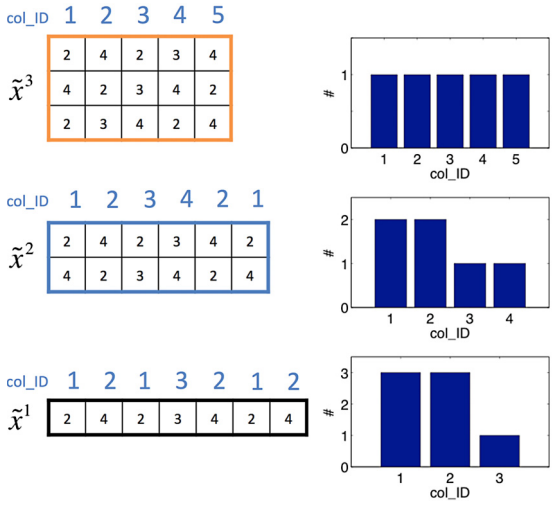


Fig. 8. Trajectory embedding vectors with assigned IDs (left) and corresponding histograms (right).

Of course we have calculated the approximation inefficiently by employing the recurrence plot $\widetilde{\mathbf{R}}$. Using Theorem 2 and Algorithm 1 we may compute $\widetilde{DET}^{(2)}$ algorithmically.

Following Algorithm 1 we assign unique identifiers to the rows of $\tilde{\vec{x}}^{\nu}$. Note that in Fig. 8 the $\vec{x}^{\nu}$ are transposed, hence in this case we are interested in unique columns. The histograms of the identifiers (col_ID) are charged and due to Theorem 1 we calculate (compare with Fig. 8)

- $\widetilde{\mathcal{PP}}^{(1)} = 3^2 + 3^2 + 1^2 = 19$
- $\widetilde{\mathcal{PP}}^{(2)} = 2^2 + 2^2 + 1^2 + 1^2 = 10$
- $\widetilde{\mathcal{PP}}^{(3)} = 1^2 + 1^2 + 1^2 + 1^2 + 1^2 = 5$

Finally, using Definition 1 (which is based on Theorem 2) we get the same result as before in Eq. (20):

$$\widetilde{DET}^{(2)} = \frac{2 \cdot \widetilde{\mathcal{PP}}^{(2)} - (2-1) \cdot \widetilde{\mathcal{PP}}^{(3)}}{\widetilde{\mathcal{PP}}^{(1)}} = \frac{15}{19} \approx 0.79.$$

References

[1] N. Marwan, M.C. Romano, M. Thiel, J. Kurths, Recurrence plots for the analysis of complex systems, Phys. Rep. 438 (5–6) (2007) 237–329, http://dx.doi.org/10.1016/j.physrep.2006.11.001.
[2] C.L. Webber Jr., N. Marwan, Recurrence Quantification Analysis – Theory and Best Practices, Springer, Heidelberg, 2015.
[3] C.L. Webber Jr., N. Marwan, A. Facchini, A. Giuliani, Simpler methods do it better: success of recurrence quantification analysis as a general purpose data analysis tool, Phys. Lett. A 373 (2009) 3753–3756, http://dx.doi.org/10.1016/j.physleta.2009.08.052.
[4] N. Marwan, J. Kurths, Comment on "Stochastic analysis of recurrence plots with applications to the detection of deterministic signals" by Rohde et al. [Physica D 237 (2008) 619–629], Physica D 238 (16) (2009) 1711–1715, http://dx.doi.org/10.1016/j.physd.2009.04.018.
[5] A. Facchini, C. Mocenni, N. Marwan, A. Vicino, E.B.P. Tiezzi, Nonlinear time series analysis of dissolved oxygen in the Orbetello Lagoon (Italy), Ecol. Model. 203 (3–4) (2007) 339–348, http://dx.doi.org/10.1016/j.ecolmodel.2006.12.001.
[6] N. Marwan, S. Schinkel, J. Kurths, Recurrence plots 25 years later – gaining confidence in dynamical transitions, Europhys. Lett. 101 (2013) 20007, http://dx.doi.org/10.1209/0295-5075/101/20007.
[7] S. Spiegel, J.-B. Jain, S. Albayrak, A Recurrence Plot-Based Distance Measure, Springer, Cham, 2014, pp. 1–15.
[8] Serrà, M. Müller, P. Grosche, J.L. Arcos, Unsupervised music structure annotation by time series structure features and segment similarity, IEEE Trans. Multimed. 16 (5) (2014) 1229–1240, http://dx.doi.org/10.1109/TMM.2014.2310701.
[9] S. Raiesdana, S.M.R.H. Golpayegani, S.M.P. Firoozabadi, J.M. Habibabadi, On the discrimination of patho-physiological states in epilepsy by means of dynamical measures, Comput. Biol. Med. 39 (12) (2009) 1073–1082, http://dx.doi.org/10.1016/j.compbiomed.2009.09.001.
[10] J.M. Nichols, S.T. Trickey, M. Seaver, Damage detection using multivariate recurrence quantification analysis, Mech. Syst. Signal Process. 20 (2) (2006) 421–437, http://dx.doi.org/10.1016/j.ymssp.2004.08.007.
[11] T. Rawald, M. Sips, N. Marwan, D. Dransch, Fast Computation of Recurrences in Long Time Series, Springer, Cham, 2014, pp. 17–29.
[12] K. Kulkarni, P. Turaga, Recurrence textures for human activity recognition from compressive cameras, 2012, pp. 1417–1420, http://dx.doi.org/10.1109/ICIP.2012.6467135.
[13] G. Varni, G. Dubus, S. Oksanen, G. Volpe, M. Fabiani, R. Bresin, J. Kleimola, V. Välimäki, A. Camurri, Interactive sonification of synchronisation of motoric behavior in social active listening to music with mobile devices, J. Multimodal User Interfaces 5 (3–4) (2012) 157–173, http://dx.doi.org/10.1007/s12193-011-0079-z.
[14] M. Morrell, Brain stimulation for epilepsy: can scheduled or responsive neurostimulation stop seizures?, Curr. Opin. Neurol. 19 (2006) 164–168, http://dx.doi.org/10.1097/01.wco.0000218233.60217.84.
[15] F. Mormann, R.G. Andrzejak, C.E. Elger, K. Lehnertz, Seizure prediction: the long and winding road, Brain 130 (2007) 314–333, http://dx.doi.org/10.1093/brain/awl241.
[16] N. Thomasson, T.J. Hoeppner, C.L. Webber Jr., J.P. Zbilut, Recurrence quantification in epileptic EEGs, Phys. Lett. A 279 (1–2) (2001) 94–101, http://dx.doi.org/10.1016/S0375-9601(00)00815-X.
[17] W. Zhang, G. Worrell, B. He, Recurrence based deterministic trends in EEG records of epilepsy patients, 2008, pp. 391–394, http://dx.doi.org/10.1109/ITAB.2008.4570657.
[18] T. Zhu, L. Huang, S. Zhang, Y. Huang, Predicting epileptic seizure by recurrence quantification analysis of single-channel EEG, in: Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues, in: Lecture Notes in Computer Science, vol. 5226, 2008, pp. 438–445.
[19] U.R. Acharya, S. Vinitha Sree, S. Chattopadhyay, W. Yu, P.C.A. Ang, Application of recurrence quantification analysis for the automated identification of epileptic EEG signals, Int. J. Neural Syst. 21 (3) (2011) 199–211, http://dx.doi.org/10.1142/S0129065711002808.
[20] N. Marwan, N. Wessel, U. Meyerfeldt, A. Schirdewan, J. Kurths, Recurrence plot based measures of complexity and its application to heart rate variability data, Phys. Rev. E 66 (2) (2002) 026702, http://dx.doi.org/10.1103/PhysRevE.66.026702.

[21] A. Trzebski, M. Śmietanowski, Non-linear dynamics of cardiovascular system in humans exposed to repetitive apneas modeling obstructive sleep apnea: aggregated time series data analysis, Autonom. Neurosci. 90 (1–2) (2001) 106–115, http://dx.doi.org/10.1016/S1566-0702(01)00275-2.
[22] T. Rybak, Using GPU to improve performance of calculating recurrence plot, Zeszyty Nauk. Politech. Białostockiej Inform. 6 (2010) 77–94, http://www.bogomips.w.tkb.pl/publications/zn-2010-cuda.pdf.
[23] S. Spiegel, S. Albayrak, An order-invariant time series distance measure – position on recent developments in time series analysis, in: Proceedings of 4th International Conference on Knowledge Discovery and Information Retrieval (KDIR), SciTePress, 2012, pp. 264–268.
[24] S. Spiegel, J.-B. Jain, S. Albayrak, A recurrence plot-based distance measure, in: N. Marwan, M. Riley, A. Giuliani, C.L. Webber Jr. (Eds.), Translational Recurrences, in: Springer Proceedings in Mathematics & Statistics, vol. 103, Springer International Publishing, 2014, pp. 1–15.
[25] N.H. Packard, J.P. Crutchfield, J.D. Farmer, R.S. Shaw, Geometry from a time series, Phys. Rev. Lett. 45 (9) (1980) 712–716, http://dx.doi.org/10.1103/PhysRevLett.45.712.
[26] N. Marwan, A historical review of recurrence plots, Eur. Phys. J. Spec. Top. 164 (1) (2008) 3–12, http://dx.doi.org/10.1140/epjst/e2008-00829-1.
[27] C. Bandt, A. Groth, N. Marwan, M.C. Romano, M. Thiel, M. Rosenblum, J. Kurths, Analysis of bivariate coupling by means of recurrence, in: Understanding Complex Systems, Springer, Berlin, Heidelberg, 2008, pp. 153–182.
[28] I.P.P. Grassberger, Estimation of the Kolmogorov entropy from a chaotic signal, Phys. Rev. A 9 (1–2) (1983) 2591–2593.
[29] J. Wiedermann, The complexity of lexicographic sorting and searching, in: J. Bečvář (Ed.), Mathematical Foundations of Computer Science 1979, in: Lecture Notes in Computer Science, vol. 74, Springer, Berlin, Heidelberg, 1979, pp. 517–522.
[30] S. Schinkel, O. Dimigen, N. Marwan, Selection of recurrence threshold for signal detection, Eur. Phys. J. Spec. Top. 164 (1) (2008) 45–53, http://dx.doi.org/10.1140/epjst/e2008-00833-5.
[31] L.L. Trulla, A. Giuliani, J.P. Zbilut, C.L. Webber Jr., Recurrence quantification analysis of the logistic equation with transients, Phys. Lett. A 223 (4) (1996) 255–260, http://dx.doi.org/10.1016/S0375-9601(96)00741-4.