Chaos, Solitons & Fractals 51 (2013) 13–21

Contents lists available at SciVerse ScienceDirect

Chaos, Solitons & Fractals

Nonlinear Science, and Nonequilibrium and Complex Phenomena

journal homepage: www.elsevier.com/locate/chaos

Evaluating Lyapunov exponent spectra with neural networks

A. Maus [a,*], J.C. Sprott [b]

[a] Computer Sciences Department, University of Wisconsin, 1513 University Avenue, Madison, WI 53706, United States
[b] Physics Department, University of Wisconsin, 1150 University Avenue, Madison, WI 53706, United States

ARTICLE INFO

Article history:
Received 12 June 2012
Accepted 6 March 2013
Available online 9 April 2013

ABSTRACT

A method using discrete cross-correlation for identifying and removing spurious Lyapunov exponents when embedding experimental data in a dimension greater than the original system is introduced. The method uses a distribution of calculated exponent values produced by modeling a single time series many times or multiple instances of a time series. For this task, global models are shown to compare favorably to local models traditionally used for time series taken from the Hénon map and delayed Hénon map, especially when the time series are short or contaminated by noise. An additional merit of global modeling is its ability to estimate the dynamical and geometrical properties of the original system such as the attractor dimension, entropy, and lag space, although consideration must be taken for the time it takes to train the global models.

1. Introduction

When presented with an experimental time series from a dynamical system, one is often faced with the question of whether the underlying dynamic is chaotic, and if so to quantify the sensitive dependence on initial conditions and attractor dimension. Such information is contained in the spectrum of Lyapunov exponents (LEs) which measures the rate of divergence (or convergence if negative) of nearby trajectories in the state space. Lyapunov exponents describe the evolution of a ball of initial conditions as they are stretched and squashed into an ellipsoid with principal axes corresponding to each exponent [1]. By convention, the LE spectrum is ordered from greatest to least values, and it can be used to calculate important measures of the dynamical system. For example, the Kolmogorov-Sinai entropy can be determined by summing the positive LEs according to Pesin's identity [2], the attractor dimension can be estimated from the LE spectrum using the Kaplan–Yorke conjecture [3], and the state space dimension can be determined from the number of non-spurious LEs.

If one knows the equations that produced the data, the LE spectrum can be calculated exactly using the standard Jacobian algorithm in which the eigenvalues are calculated for the product of the Jacobian matrices along the orbit [4]. Gram-Schmidt re-orthonormalization is used to factor out multipliers that lead to numerical divergence and performs row reduction, helping to retain product matrix column independence. Even so, it is often necessary to follow the orbit for a long time to ensure that the state space is adequately sampled and that the values obtained by the method have converged.

Since one rarely has the luxury of knowing the equations, methods for approximating the LEs directly from the data have been proposed [4–8]. These methods typically estimate the local Jacobian by fitting the data in the vicinity of each data point to a simple function by considering the local neighborhood around a point. Since each neighborhood must contain sufficiently many points, local methods are severely limited by the length of the time series and by the inevitable noise that accompanies experimental data, but they are reasonably fast since they require only a single pass through the data.

An alternative is to construct a global model of the data, which is more difficult but has the advantage of using the entire data set and better averages the noise present in the data. Training such a global model with a sufficiently

* Corresponding author. Tel.: +1 6088901445; fax: +1 6088901438.
E-mail addresses: amaus@a-ma.us (A. Maus), sprott@physics.wisc.edu (J.C. Sprott).

0960-0779/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.chaos.2013.03.001

general functional form allows approximation of the LE spectrum using the standard Jacobian algorithm as shown in [9].

With either method, one must choose an embedding dimension that adequately describes the state space or dynamics of the system. Takens [10] showed that complete unfolding is guaranteed for a D-dimensional system if the model is embedded in a dimension greater than $2D$. Algorithms such as false nearest neighbors and the plateau of the correlation dimension calculation can be used to determine the optimal embedding dimension [11,12]. We have recently proposed a method in which a global model of the data is used to determine the optimal embedding dimension, and more particularly, the lag space for cases in which only a subset of the embedding dimension is active [13].

If the embedding dimension d of a model is higher than the system dimension D, the estimated LE spectrum will contain $d - D$ spurious LEs [14]. Embedding in a dimension higher than necessary allows multiple solutions with different dynamics that have the same projection onto the D-dimensional manifold. Ideally, the dynamics in dimensions greater than D would be limited. However, these exponents are often positive and greater than the largest true LE even when the data are generated from simple chaotic maps [15] and independent of any measurement or reconstruction function used [16]. Identification and removal of these spurious exponents is crucial since inaccuracies in the spectrum lead to incorrect estimation of other measures.

2. Spurious Lyapunov exponent identification

Methods have been proposed to identify spurious LEs with varying degrees of success. Brown et al. [5] recommended adding white measurement noise to the data. True Lyapunov exponents tend to be significantly more resistant to noise than spurious ones since dimensions unrelated to the dynamics of the time series are dominated by noise and vary with it, while the other dimensions are simply contaminated. Alternately, varying the embedding dimension may identify spurious exponents since true exponents are independent of the choice of embedding while spurious exponents will continually shift [17].

Parlitz [18] proposed using time reversal to identify spurious LEs. Reversing the order of the data points in the time series results in a dynamic where even dimensions are reflected and odd dimensions are rotated in time-delayed embedding space. Such reversal generally converts attractors into repellors, reversing the sign of each LE. Models tend to fit the time-reversed data just as well as the actual data since the same state space points, outliers, noise, and local curvature are present. Models that fit these data well will have true LEs whose values change sign when time is reversed while spurious exponents do not reverse sign since those exponents arise from fitting a function to dimensions in which no information is available from the time series.

As an extension of Parlitz's method, we propose a method based on time series reversal that uses discrete cross-correlation, referred to as cross-correlation for the rest of the article, to identify and remove spurious exponents. Two sets of calculated LE values are produced from modeling a time series and its time-reversed counterpart many times. After inverting the LE values in the time-reversed set, a probability histogram is created for each set by sorting the LE values into bins in ascending order and counting the number of values in each bin then normalizing the resulting vector. Discrete cross-correlation [19] is applied to the two histograms. The forward histogram is slid through the time-reversed histogram to create a histogram that is then normalized. Spurious exponents are identified and removed by identifying peaks in this new histogram. To estimate the LE spectrum values, one can either take the center of the highest probability bins as the LE value or fit a parabola to the three nearest bin centers and take the maximum of the parabola as the LE value. The latter was used to produce the results in this study. For higher-dimensional systems with LE values that cluster closely together, greater resolution or more bins may be required to resolve overlaps in the values. The results of cross-correlation for a single sample of 512 points from the Hénon map are shown in Fig. 1. In Fig. 1(A), a histogram of LE values, with 640 bins and 36,138 training instances with $d = 6$ is shown for the forward time series. In this graph we expect the models of the data to find LE values near the true LE values, 0.422 and −1.623, indicated by the arrows on the horizontal axis below Fig. 1(C), and the histogram shows high probability bins or peaks near these values. Without time reversal, the spurious exponents are spread out mostly in the range between −2 and −1 where a true exponent, −1.623, is known to exist. In Fig. 1(B), the histogram of LE values for 34,918 training instances is shown for the time-reversed time series with the sign of the LE values already inverted and Fig. 1(C) shows the results of cross-correlation.

The cross-correlation method is accurate in removing spurious LEs since it averages noise within the LE values between different training instances. Since the method is general, the model used to reconstruct the LE spectra can be easily changed. However, the method has particular issues that are time-series and model dependent. The method requires stochasticity in the model, training algorithm, or data that allow variation in the calculated LE values which is highlighted in Fig. 2 where local linear fits are used to estimate the Hénon map's LE spectrum. In Fig. 2(A) the forward and time-reversed histograms are shown overlaid with only a few bins overlapping, leading to inaccurate LEs in Fig. 2(B). This problem stems from using the same model parameters for multiple training instances with a purely deterministic model. For parametric models with only a few parameters such as local linear fits, one can vary the model parameters to obtain variations in the estimated LEs. Models such as neural networks may also have variable parameters that are learned through a stochastic training algorithm. For example, the training algorithm used in this paper depends on randomly selecting the weights while trying to improve the model's accuracy. Through this random selection, the model searches a large space of possible solutions and will rarely find the exact same solution and LE spectrum. Backpropagation, another training algorithm, may produce the

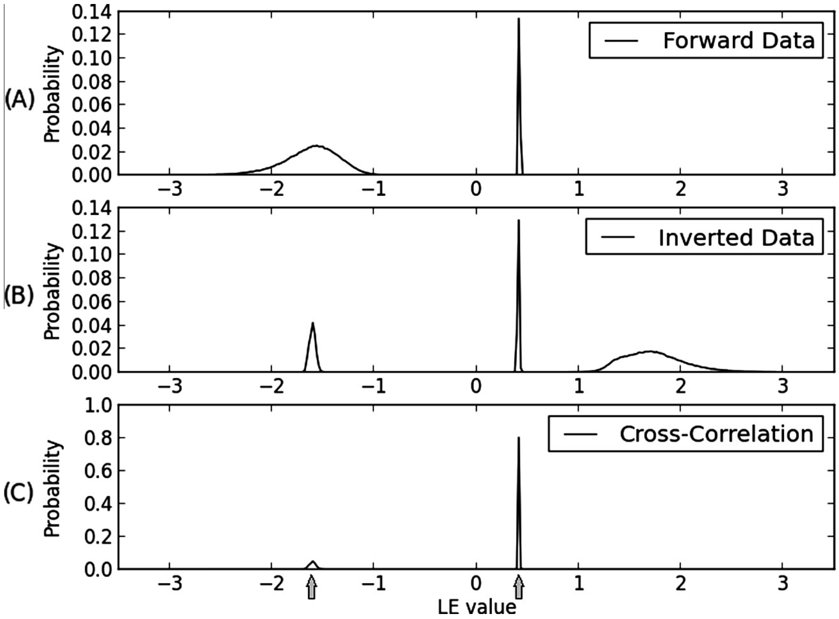

Fig. 1. Cross-correlating the probability distributions produced by training many neural networks with $d = 6$ on the Hénon map, with 512 points, removes the spurious Lyapunov exponents. The spurious exponents in the forward time series are removed around the negative exponent value because they shift when the time series is reversed and modeled. The peaks in the (C) correspond to exponent values, 0.413 and −1.602, which are close to the actual values, 0.422 and −1.623, indicated by the arrows.

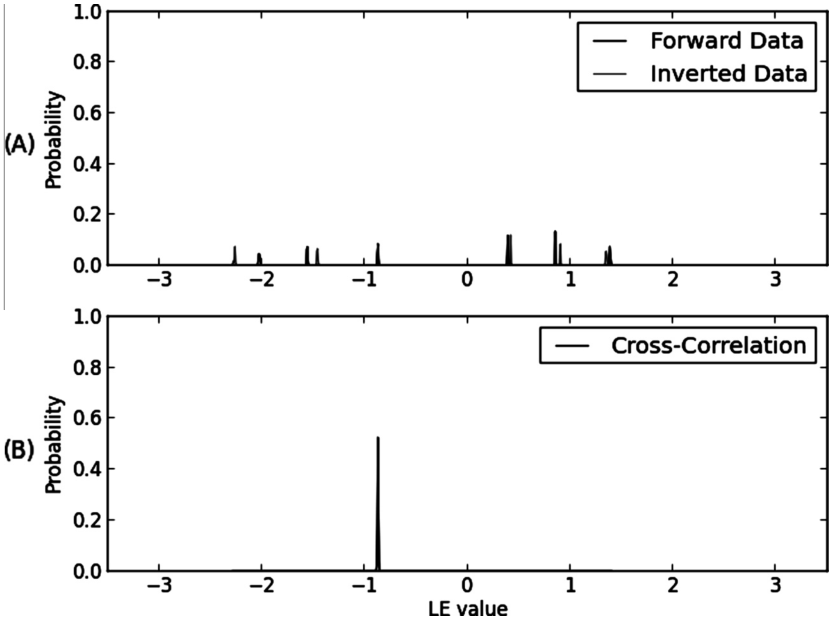

Fig. 2. The LE value probability distributions and cross-correlation for Hénon map time series with 32,768 points produced by local linear models with 150 neighbors. LE values tend to have low variance with an increased data length. The resulting cross-correlation is problematic if the distributions from the forward and time-reversed exponent values do not overlap. Introducing stochasticity by incorporating noise into the data, LE, or neighborhood may increase the amount by which values overlap.

exact same LE values for the same data set since it usually only searches in directions that improve accuracy. However, variations can be introduced by randomly initializing the neural network weights or by perturbing the weights as the network is trained. As an alternative to varying the model parameters, one could vary the dataset through the addition of noise or by varying how the data is partitioned between training instances.

Cross-correlation in instances where an overspecified model leads to highly variable LE values or numerous peaks in the cross-correlated graph results in more difficult spurious LE removal. A different method for binning the data can be chosen to overcome the increased variability by giving more resolution to regions of the histogram that have higher density while keeping the number of bins consistent between the forward and time-reversed time series. To do this, one can combine the forward and time-reversed LE values into one histogram and hold the number of points per bin fixed to obtain bins with a set number of points. Using the bin boundaries, one separates the histograms and performs cross-correlation to record the highest probability estimated LEs. This process could be repeated with a different number of points per bin to determine which LE values are most consistent. Binning based on density also has a smoothing effect on the data and may reduce inaccuracies caused by binning too large of a region. Using either strategy or combining both results can lead to more accurate LE estimates.

3. The models

Neural network and local linear fits were used to estimate LE spectra for discrete-time systems. The models were trained on time-delayed data taken from the Hénon map and the delayed Hénon map and optimized for next-step prediction based on d dimensions or time lags. In addition to removing the spurious exponents, the models produce more accurate LE values when using cross-correlation than by simply averaging a number of trials. The advantages of using global models can be seen in each of the experiments.

As one of the global models considered, neural networks have a rich history. Hornik et al. [20] proved that neural networks are universal approximators, showing that any smooth function could be represented to arbitrary accuracy by a single-layer feed-forward neural network with sufficiently many neurons. Single-layer neural networks are composed of a matrix of coefficients that represent input connection strengths to each neuron and a vector representing the strength of each neuron's respective contribution to the output, shown schematically in Fig. 3.

The general form used in this study is

$$\hat{x}_k = \sum_{i=1}^{n} b_i \tanh\left(a_{i0} + \sum_{j=1}^{d} a_{ij} x_{k-j}\right)$$

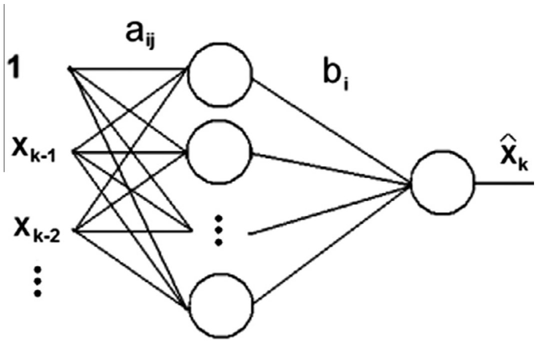

Fig. 3. The neural network schematic.

where n is the number of neurons, a_{ij} is an n by d matrix of coefficients, b_i is a vector of coefficients of length n, x is the training time series, and $\hat{x}_k$ is the predicted value for time step k. While these neural networks used a hyperbolic tangent nonlinearity, there is nothing to prevent using other functions such as polynomials. Using a quadratic nonlinearity with only two neurons allows a nearly perfect fit to Hénon map data, but we consider this to be a trivial and not very useful example other than to check the calculation.

One can fit a neural network to data by adjusting the connection strengths a and b to minimize the error. To avoid having the network always train to one or more identical solutions, a stochastic training method was used. The method resembles simulated annealing, with the coefficients chosen randomly from a slowly shrinking Gaussian neighborhood of the current best solution. The Gaussian is taken to have an initial standard deviation of 2^j centered on zero to give preference to the most recent time lags (small j values). The connection strengths are chosen to minimize the average one-step mean-square prediction error:

$$e = \frac{\sum_{k=d+1}^{c} (\hat{x}_k - x_k)^2}{c - d}$$

where c is the length of the time series (the number of data points). For each of the systems described, a network with four neurons ($n = 4$) was trained for one million iterations on the data. Adding more neurons allows better fits but at the expense of computation time. With an unknown system, it may be necessary to train with different numbers of neurons to identify the optimum network size.

The neural network LE spectra estimations are compared to those obtained using the publicly available TISEAN package which implements Sano and Sawada's algorithm for estimating the LE spectra of experimental time series [8,21]. The LE spectrum in TISEAN is estimated by fitting local linear models of the form $s_{n+1} = a_n\ s_n + b_n$ where a_n and b_n are given by the least squares minimization of $\sigma^2 = \sum_l (s_{l+1} - a_n s_l - b_n)^2$ and $\{s_l\}$ is the set of neighbors that are within some ball of distance ϵ of s_n or some fixed number of nearest neighbors of s_n in the d-dimensional embedding. After approximating the linearized flow map, a_n, which ends up being the first row of the local Jacobian, the LE spectrum is estimated using the standard Jacobian algorithm.

We compare the global neural network model to local linear fits by testing their ability to approximate the Hénon map and delayed Hénon map systems. One advantage of global models is their ability to create an analytical function of the experimental data. The hope is that if the function fits the data accurately, then one can generate an infinite amount of data from the model and estimate dynamical and topological properties of the original system that would traditionally require a much longer data record. In practice this is not generally the case except for simple systems. However, if the global model has been well trained, it can represent data even in regions of state space where data are sparse. Additionally, global models tend to be robust to small amounts of noise due to their ability to average this noise over the entire attractor.

Although global models have these advantages, they often suffer from training inefficiency. For example, a neural network must be trained to fit the data, requiring many passes through the data set and many changes to the connection strengths. Furthermore, for many training algorithms including the one described, there is no proof of convergence or guarantee that the neural network will ever find the global solution since the space being explored is nonconvex. Unlike neural networks, local linear fits excel in their simplicity and efficiency. Local models need only one pass through the data to calculate averages and nearest neighbors and to estimate the LE spectrum. However, local models suffer when the time series is short or noisy, as will be shown. They also do not accurately represent regions of the attractor where data are sparse or regions off the attractor.

4. Numerical results

The advantages of neural networks are illustrated by comparing their predicted Lyapunov exponent spectra with the spectra predicted from the local linear model using data from the Hénon map [22] and data of various lengths taken from the delayed Hénon map [23] with and without added noise.

The Hénon map in time-delayed form,

$$x_k = 1 - 1.4x_{k-1}^2 + 0.3x_{k-2}$$

and its strange attractor in Fig. 4, represents a simple two-dimensional system with LE values of 0.422 and −1.623 as shown in [24]. For the following results, the embedding dimension of the models was set to six; since the Hénon map is a two-dimensional system, four spurious LEs should be produced. The number of nearest neighbors used in the local linear fits was fixed at 230; a value chosen to minimize the LE spectrum error using a tuning set of 4096 points taken from the Hénon map. The number of neurons needed to model the data by the neural network can also be chosen in a similar manner since the use of a training set helps to estimate the model complexity required for the system. Neural networks with four neurons can accurately model the systems studied.

Table 1 summarizes the results for each method applied to the Hénon map with 32,768 points where 200 trials represents the models trained on 100 forward time series and 100 time reversed time series. Even for this simple map, spurious exponents are intermingled with real values and in some cases are within just a few percent of the actual exponents. Cross-correlation serves to remove these spurious exponents but at the expense of accuracy. Since we know the actual LE values of the Hénon map, we can compare the averaged exponents to the actual exponents for the neural network by manually removing the spurious exponents from the averaged forward exponents, resulting in absolute errors of 0.007 and 0.068 while cross-correlation removes the spurious exponents and has absolute errors of 0.007 and 0.041.

The local linear fits were able to estimate a positive exponent with an absolute error of 0.009 after cross-correlation but could not determine the negative exponent accurately. One of the difficulties in determining the LE spectrum is estimating the negative exponent which measures the convergence of orbits to the attractor, for which data are usually absent in the time series. For both the forward and reversed time series, spurious exponents are intermingled with the actual exponents and in the forward time series, a spurious exponent is produced that is larger than the largest exponent. Even with cross-correlation, two spurious exponents are produced, highlighting the fact that unless local linear fits find LE values consistent between the forward and reversed time series, the distributions of LE values may not overlap and produce any cross-correlated LE values. Local linear fits tend to produce

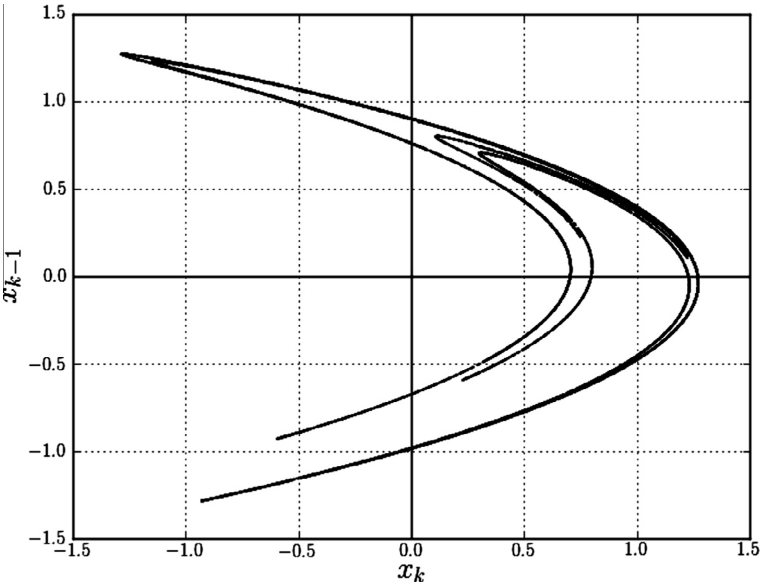

Fig. 4. The strange attractor of the Hénon map.

Table 1
The averaged exponents for 100 trials with 32,768 points in each time series taken from the Hénon map. Also included for each model are the results of cross-correlating the 200 time series. LEs are removed from the results if they are considered to be spurious, the absolute error is greater than 0.5 when compared to the actual values. In practice, greater accuracy could be achieved by varying data length, noise levels, or model parameters.

		λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
Actual exponents		0.422	−1.623				
Neural network with four neurons	Data	0.415	−1.399	−1.555	−1.703	−1.829	−2.291
	Reversed	1.565	−0.434	−1.445	−1.540	−1.704	−1.890
	Cross-correlation	0.415	−1.582				
	Error	0.007	0.041				
Local linear fits with 230 neighbors	Data	1.318	0.809	0.398	−0.790	−1.477	−1.904
	Reversed	2.099	1.326	0.801	−0.429	−0.869	−1.340
	Cross-correlation	1.330	0.413	−0.796			
	Error		0.009				

low variance LE values when trained on a low complexity system such as the Hénon map with a large number of points as seen in Fig. 2. By varying the number of neighbors, additive noise, and number of points in the time series, values may overlap and produce more accurate cross-correlated LE values.

The Hénon map can be generalized by replacing the linear term with an earlier iterate in the time series [23]. The delayed Hénon map,

$$x_k = 1 - 1.6x_{k-1}^2 + 0.1x_{k-D}$$

whose attractor is shown in Fig. 5, has an adjustable dimension parameter D that determines its complexity. For these results, D is taken to be four, requiring an embedding dimension of four. Related to the embedding dimension, its lag space has only two dimensions, 1 and D, since the dynamics of the time series depend only on the first and D-th time delay [13,25]. The LE spectrum for this discrete dynamical system as determined from the equations using Wolf's method [4] is 0.381, −0.813,−0.900, −0.971. The values are believed to be accurate to three significant digits because the exponents have converged to these values after one billion iterations from ten different initial conditions within the basin of attraction.

If one embeds the time series in a six-dimensional space, two spurious LEs will be produced since the original system only requires an embedding of four. Table 2 shows how neural networks compare to local linear fits with 10 neighbors when each method is used to estimate the spectrum of the delayed Hénon map.

Similar to the Hénon map, both of the models produce two spurious exponents that are dissimilar from the rest. The neural network's spurious exponents are highly negative compared to the real exponents, making them relatively easy to identify and remove for this map. Local linear fits produce spurious exponents that are intermingled with the real exponents. If we only select the averaged exponents nearest to the actual exponent values in the forward time series, the exponent values have a lower error than the reversed data. Using cross-correlation increases the overall accuracy of the spectrum for the neural network however local linear fits only produce one accurate positive exponent and a spurious exponent

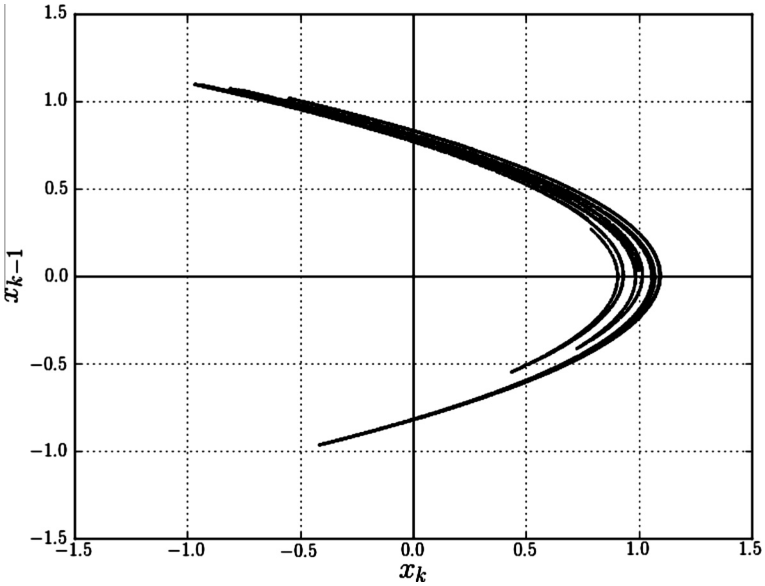

Fig. 5. The strange attractor of the delayed Hénon map ($D = 4$).

Table 2
The averaged exponents and resulting cross-correlation for 100 trials each with 32,768 points taken from the delayed Hénon map with $D = 4$. 1% White noise was introduced into the results of the local linear fits during cross-correlation so LE values in the histograms would overlap.

		λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
Actual exponents		0.381	−0.813	−0.900	−0.971		
Neural network with 4 neurons	Data	0.381	−0.794	−0.899	−0.988	−2.524	−2.906
	Reversed	1.042	0.763	0.441	−0.482	−2.611	−3.035
	Cross-correlation	0.383	−0.765	−0.903	−0.964		
	Error	0.002	0.048	0.003	0.007		
Local linear fits with 10 neighbors	Data	0.926	0.393	−0.292	−0.829	−0.973	−1.199
	Reversed	1.128	0.920	0.755	0.225	−0.407	−0.907
	Cross-correlation	0.9207	0.40				
	Error		0.015				

that is larger than the most positive real LE. This inaccuracy arises from the lack of variation in LE values even with 1% white noise introduced into the resulting LE values to increase overlaps in the histograms.

In practice, one rarely has the advantage of having long, noise-free data sets such as the ones described for the Hénon and delayed Hénon maps. To test and compare the performance of each method under more adverse conditions, variable noise and data length were introduced to time series from the delayed Hénon map. To test data length, time series were collected with 64 points through 32,768 points in powers of two. For each length of data, distributions of LE values were produced from two hundred different models embedded in six dimensions, one hundred on the forward time series and one hundred on the time-reversed time series. Local linear fits were parameterized with 40 neighbors and neural networks with four neurons. After cross-correlation, the mean square LE error was calculated between the estimated LE values and the expected LE values as shown in Fig. 6. The results show that the mean square errors for the neural network are about ten times smaller than the local linear fits in all of these time series. However, the error of the local linear models decreases as more points are added, whereas the global model errors change very little. The increase in error when the number of points reaches 32,768 may indicate that the neural networks need to be trained further.

The methods were also compared on time series with added measurement noise. The models were embedded in six dimensions and trained on varying levels of Gaussian white noise added to 4096 points taken from the delayed Hénon map. The error in the average values of the exponents for one hundred trials are shown in Fig. 7. The neural network outperforms the local linear fits for all levels of noise shown. For adequately sampled attractors, the models successfully average small levels of measurement noise; however, neural networks' robustness lies in their ability to create a global deterministic model that is noise free albeit with some distortion of the signal.

The models were finally tested on normalized data taken from a nuclear magnetic resonance laser [26]. We apply the cross-correlation method to local linear fits and neural network models with $d = 3$ trained on this data set. The neural network was initialized with four neurons,

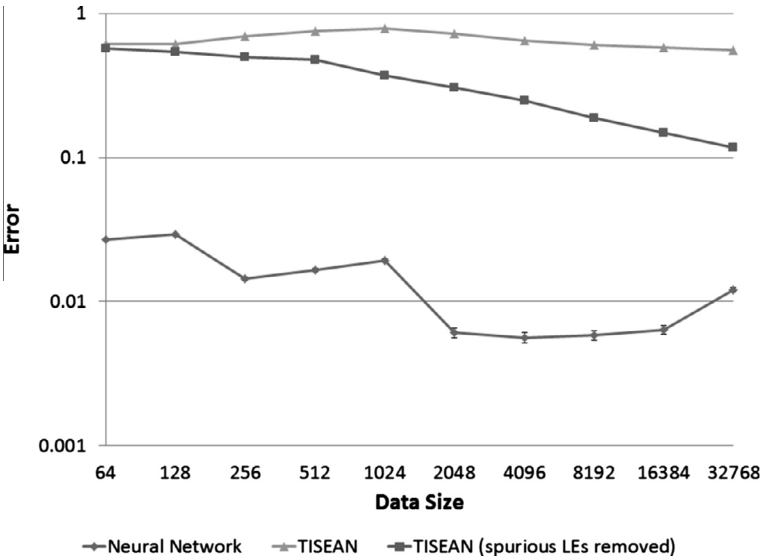

Fig. 6. The error in the averaged exponents for 100 trials from each model for the delayed Hénon map ($D = 4$) for varying length time series. The removal of spurious LEs for local linear fits was performed manually.

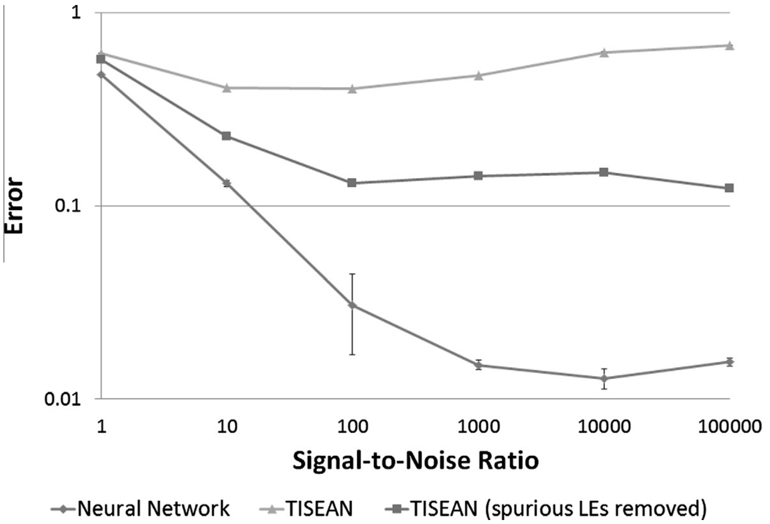

Fig. 7. The error in the averaged exponents for 100 trials from each model trained on 4096 points from the delayed Hénon map ($D = 4$) for time series corrupted with Gaussian white noise. The error bars represent one standard deviation. The removal of spurious LEs for local linear fits was performed manually.

Table 3
The averaged exponents and resulting cross-correlation for 200 trials of the nuclear magnetic resonance laser data set.

		λ_1	λ_2	λ_3
Neural network with four neurons	Data	0.251	−0.315	−1.178
	Reversed	0.197	−0.247	−0.604
	Cross-correlation	0.290	−0.086	−0.364
Local linear fits with 50 neighbors	Data	0.267	−0.652	−1.060
	Reversed	0.415	−0.293	−1.011
	Cross-correlation	0.277		

a value chosen by minimizing the next step prediction error on a training set of 1000 points and trained for one million iterations. The neural network then estimated the LE spectrum using 25,000 points and 1% noise. The number of neighbors used in local linear fits was set to 50, a value chosen by minimizing the next step prediction error on 15,000 points. Local linear fits used blocks of 25,000 points and 1% noise to introduce the variation needed in cross-correlation. The results for 200 trials for each model can be seen in Table 3. For the forward time series, the neural networks produce three exponents with two exponent values that are similar to local linear fits. Since both models produce exponent values near 0.28, one could study how consistent the exponent values are by varying the time series further and studying its effect. In [21], a similar data set with LE values more consistent with the local linear fits forward time series averaged exponent values is studied.

5. Conclusions

Neural network and local linear models are fit to the Hénon map and its more complex counterpart, the delayed Hénon map. Results are also shown for time series with various lengths of data and additive noise. Local linear fits are shown to work well for time series with many thousands of data points while global neural network models produce accurate LE spectra in these simple systems with as few as 64 points. Local linear fits rely on simple local models to determine the LE spectrum but require a large number of points so the neighborhood of each data point is adequately populated. When noise is introduced, local linear fits and neural networks fail in cases where the noise is prevalent, although neural networks are shown to be more robust to noise, highlighting one of the strengths of global models. Neural networks build a global model of the data, but there is a trade-off between the amount of computation required and accuracy of the model. In computation-sensitive applications, local linear fits would be advantageous since the LE spectrum can be estimated in one pass through the data.

A method for removing spurious exponents is introduced that can estimate the actual exponent values by cross-correlating the distributions of LE values produced by estimating the LE spectrum for a time series and its time-reversed counterpart many times. The method is shown to perform favorably for the systems studied and works with any model of the time series, but care should be taken when LEs have very similar values within a given system's LE spectrum or when the forward and reversed LE value distributions in the resulting cross-correlation do not overlap.

The use of neural networks and other models remains important in the study of dynamical systems, and much can be gleaned by analyzing models trained on experimental data. Further studies involving transformations of the data through various observer functions and the study of continuous-time systems are beyond the scope of this paper but will be the subject of future work.

A Windows version of the program used in this study is available on the Web at http://sprott.physics.wisc.edu/chaos/maus/lagspace.htm.

References

[1] Oseledec VI. Multiplicative ergodic theorem: characteristic Lyapunov exponents of dynamical systems. Trans Moscow Math Soc 1968;19:179210.
[2] Pesin YB. Characteristic Lyapunov exponents and smooth ergodic theory. Russ Math Surv 1977;32.
[3] Kaplan J, Yorke J. Chaotic behavior of multidimensional difference equations. In: Peitgen HO, Walther HO, editors. Functional differential equations and approximation of fixed points. Lecture notes in mathematics, vol. 730. Berlin/Heidelberg: Springer; 1979. p. 204–27.
[4] Wolf A, Swift JB, Swinney HL, Vastano JA. Determining Lyapunov exponents from a time series. Phys D 1985;16(3):285–317.
[5] Brown R, Bryant P, Abarbanel HDI. Computing the Lyapunov spectrum of a dynamical system from an observed time series. Phys Rev A 1991;43:2787–806.
[6] Bryant P, Brown R, Abarbanel HDI. Lyapunov exponents from observed time series. Phys Rev Lett 1990;65:1523–6.
[7] Eckmann JP, Ruelle D. Ergodic theory of chaos and strange attractors. Rev Mod Phys 1985;57:617–56.
[8] Sano M, Sawada Y. Measurement of the Lyapunov spectrum from a chaotic time series. Phys Rev Lett 1985;55:1082–5.
[9] Dechert WD, Gençay R. The topological invariance of Lyapunov exponents in embedded dynamics. Phys D 1996;90(1–2):40–55.
[10] Takens F. Detecting strange attractors in turbulence. In: Rand D, Young LS, editors. Dynamical systems and turbulence, Warwick 1980. Lecture notes in mathematics, vol. 898. Berlin/Heidelberg: Springer; 1980. p. 366–81.
[11] Kennel MB, Brown R, Abarbanel HDI. Determining embedding dimension for phase-space reconstruction using a geometrical construction. Phys Rev A 1992;45:3403–11.
[12] Grassberger P, Procaccia I. Characterization of strange attractors. Phys Rev Lett 1983;50:346–9.
[13] Maus A, Sprott J. Neural network method for determining embedding dimension of a time series. Commun Nonlinear Sci Numer Simul 2011;16(8):3294–302.
[14] Gençay R, Dechert W. An algorithm for the n Lyapunov exponents of an n-dimensional unknown dynamical system. Phys D 1992;59(13):142–57.
[15] Gençay R, Dechert WD. The identification of spurious Lyapunov exponents in jacobian algorithms. Stud Nonlinear Dyn Econom 1996;1(3):2.
[16] Tempkin J, Yorke J. Spurious Lyapunov exponents computed from data. SIAM J Appl Dyn Syst 2007;6(2):457–74.
[17] Eiswirth M, Kruel TM, Ertl G, Schneider F. Hyperchaos in a chemical reaction. Chem Phys Lett 1992;193(4):305–10.
[18] Parlitz U. Identification of true and spurious Lyapunov exponents from time series. Int J Bifurcation Chaos 1992;2:155–65.
[19] Edelson RA, Krolik JH. The discrete correlation function – a new method for analyzing unevenly sampled variability data. Astrophys J 1988;333:646–59.
[20] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Netw 1989;2(5):359–66.
[21] Hegger R, Kantz H, Schreiber T. Practical implementation of nonlinear time series methods: The TISEAN package. Chaos 1999;9(2):413–35.
[22] Hénon M. A two-dimensional mapping with a strange attractor. Commun Math Phys 1976;50(1):69–77.
[23] Sprott JC. High-dimensional dynamics in the delayed Hénon map. Electron J Theory Phys 2006;3(19):19–35.
[24] Sprott JC. Chaos and time-series analysis. New York: Oxford; 2003.
[25] Goutte C. Lag space estimation in time series modelling. In: BW, editor. Proceedings of the 1997 intl. conference on acoustics, speech, and signal processing ICASSP 97 (Munich). IEEE; 1997. p. 3313.
[26] McSharry, P. Nonlinear dynamics and chaos workshop. Last accessed: 12/09/2012.<http://people.maths.ox.ac.uk/mcsharry/lectures/ndc/ndcworkshop.shtml>.