Neural network method for determining embedding dimension of a time series

A. Maus[1] and J. C. Sprott[2]

June 2, 2010

Keywords: embedding dimension, neural network, lag space, chaos

Abstract

A method is described for determining the optimal short-term prediction time-delay embedding dimension for a scalar time series by training an artificial neural network on the data and then determining the sensitivity of the output of the network to each time lag averaged over the data set. As a byproduct, the method identifies any intermediate time lags that do not influence the dynamics, thus permitting a possible further reduction in the required embedding dimension. The method is tested on four sample data sets and compares favorably with more conventional methods including false nearest neighbors and the 'plateau dimension' determined by saturation of the estimated correlation dimension. The proposed method is especially advantageous when the data set is small or contaminated by noise. The trained network could be used for noise reduction, forecasting, and estimating the dynamical and geometrical properties of the system that produced the data, such as the Lyapunov exponent, entropy, and attractor dimension.

[1] *E-mail address*: amaus@a-ma.us

[2] *Address*: Physics Department, University of Wisconsin, 1150 University Ave., Madison, Wisconsin, 53706, USA
Telephone: 16082634449
E-mail address: sprott@physics.wisc.edu

1 Introduction

When presented with an experimental time series from which one wants to make a forecast or determine properties of the underlying dynamical system, the starting point is usually to embed the data in a time-delayed space of suitable dimension [1]. If this embedding dimension is chosen too small, distant points on the attractor will coincide or overlap in the embedding space like the 2-dimensional shadow of a 3-dimensional object. However, if it is chosen too large, the space may be poorly sampled, noise will be more problematic, and more computation is required. Consequently, it is important to choose an optimal embedding dimension. For the purposes of this paper, the optimal embedding is the one that gives best next-step predictability. However, for other studies another 'optimal' embedding may be more suitable, such as for the estimation of various metrics like the Lyapunov exponent.

A sufficient condition for the embedding dimension was provided by Takens [2] who showed that complete unfolding is guaranteed if the time-delayed embedding space has a dimension d greater than the dimension of the original state space D by an amount $d > 2D$. Sauer *et al.* [3] later showed that under most conditions, the embedding dimension need only be greater than twice the fractal dimension of the attractor. However, it is important to recognize that for many purposes, such as estimation of the correlation dimension [4], overlaps are of little consequence if their measure is sufficiently small, and in such cases, the necessary embedding dimension may be as small as the next integer larger than the dimension of the attractor.

A closely related issue is determining the extent to which each time lag influences the future. It is easy to imagine situations in which intermediate time lags are of no consequence, for example in biological systems where seasonal, gestation, or maturation delays are present [5]. In such cases the relevant 'lag

space' may in fact be smaller than the embedding dimension estimated by conventional methods since the attractor may be negligibly thin in some of the intermediate dimensions. Therefore, the dimension of the lag space indicates the number of variables needed to model the dynamics. Methods developed for linear systems such as the autocorrelation function and autoregressive moving average (ARMA) models [6] naturally provide such information, but our interest here is in more general methods that work for nonlinear systems, and in particular, ones that exhibit chaos for which the linear methods would miserably fail.

One of the earliest methods for determining an optimal embedding dimension is to estimate the correlation dimension in increasing embeddings and to declare that the proper embedding has been found when that dimension saturates. This 'plateau dimension' by its nature is optimal for determining the true correlation dimension of the attractor, but it may not adequately remove overlaps for other purposes. Furthermore, a good plateau is often lacking with real data, especially when the data set is small and/or contaminated with noise.

A more recent and commonly used method involves calculation of false nearest neighbors [7, 8] in successively higher embedding dimensions and its many variants. Neighbors are considered false if their separation increases by more than a factor of R_T when the embedding dimension is increased from j to $j+1$, where R_T is typically taken as 15. In addition, a neighbor is considered false if its separation increases by more than a factor of A_T times the standard deviation of the data, where A_T is typically taken as 2.0 [9]. In practice, as j is increased, the fraction of neighbors that are false drops to near zero, and the dimension where this occurs is the minimum embedding dimension required to unfold the attractor.

2 Neural Network Method

In this paper, we propose an alternate method using a single-layer, feed-forward artificial neural network trained on time-delayed data and optimized for next-step prediction based on d time lags, with d chosen large enough to capture the relevant dynamics but much smaller than the number of data points in the time-series c. Analysis of the trained network then allows determination of the optimal embedding for next-step predictability, lag space, and the sensitivity of the output of the network to each time lag.

Artificial neural networks have shown great promise in modeling time-series [10, 11], nonlinear prediction [12], and the analysis of underlying features of the data [13]. Hornik, *et al.* [14] proved that neural networks are 'universal approximators' because they can represent any smooth function to arbitrary precision given sufficiently many neurons. Thus we believe the method is general and applicable to most real-world systems.

The single-layer, feed-forward network shown schematically in Figure 1 uses a layer of n hidden neurons to perform next-step prediction $\hat{x}_k$ on a scalar time-series x_k according to

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

where a_{ij} is an $n \times d$ matrix of coefficients, and b_i is a vector of length n. The a_{ij} matrix represents the connection strengths to the input of the network, and the b_i vector is used to control the contribution of each neuron to the output of the network. The vector a_{i0} is an offset that facilitates training on data whose mean is not zero.

The weights in a and b are updated using a method similar to simulated annealing in which a Gaussian neighborhood of the current best solution is ran-

domly searched, with the size of the neighborhood slowly shrinking as training progresses. In practice, 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) in the search space. The connection strengths are chosen to minimize the average one-step mean-square prediction error:

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

Once the network is trained, the sensitivity of the output to each time lag is determined by calculating the partial derivatives of the output with respect to each time lag x_{k-j} averaged over all the points in the time-series:

$$\hat{S}(j) = \frac{1}{c - j} \sum_{k=j+1}^{c} |\frac{\partial \hat{x}_k}{\partial x_{k-j}}| \tag{3}$$

For the network in Equation 1 the partial derivatives are given by

$$\frac{\partial \hat{x}_k}{\partial x_{k-j}} = \sum_{i=1}^{n} a_{ij} b_i \mathrm{sech}^2(a_{i0} + \sum_{m=1}^{d} a_{im} x_{k-m}) \tag{4}$$

The optimal embedding dimension is assumed to be the largest value of j for which $\hat{S}(j)$ has a significant value, much like the method of false nearest neighbors. The individual values of $\hat{S}(j)$ quantify the importance of each time lag, much like the terms in the autocorrelation function or the coefficients of an ARMA model for a linear system.

3 Numerical Results

The method was tested on four time-delayed chaotic maps of increasing dimension and complexity. One advantage of using data generated in this way is that the expected sensitivities to each time lag $S(j)$ can be readily determined either

by inspection of the equation that generated the data in cases where the dependence is linear or by a simple numerical averaging of the nonlinear terms over the points on the attractor using Equation 3. For these results, the attractor produced by many iterations of the trained network is visually indistinguishable from the one that produced the time series. The next-step, in-sample prediction error calculated from Equation 2 for all these noise-free cases is on the order of $e \sim 10^{-5}$. For each map, ten different instances of the time series were taken, and the neural networks' sensitivities to each time lag were calculated. This resulted in an average normalized root mean square error in the calculated sensitivities, Equation 9, less than 0.0354 with a variance less than 0.0210 for each map, with the figures representing the average of the ten trials. In addition, the finite-size Lyapunov exponent [15, 16] for the trained network is in agreement with the expected value within 5% over the range of scale sizes from 10^{-12} to 10^{-1}.

The network parameters n and d, and the number of data points c used to model the cases to be shown were chosen for their ability to produce an accurate model of the system. There is a tradeoff between accuracy and the time required to train the network. If n, d, or c are too small, the network will train poorly and give inaccurate sensitivities. For n, d, or c too large, degradation in the model will result from insufficient training or from over-fitting. Except as noted, we use the values of $n = 4$, $d = 5$, and $c = 512$, which appear adequate for the cases studied, but are not necessarily optimal for short-term prediction even for these cases. In particular, we note that the number of neurons required to get good agreement in the sensitivities is smaller than the number required for the smallest training error e, presumably because the goal is not to get the best fit to the data but only to determine the sensitivity to each time lag. In practice, one should try different values of the parameters for each time series, but we

find the method is tolerant of a wide range of choices.

The simplest example considered here is the Hénon map [17], given in time-delayed form by

$$x_k = 1 - 1.4x_{k-1}^2 + 0.3x_{k-2}, \tag{5}$$

with a strange attractor as shown in Figure 2a. From Equation 5 it is evident that this map has an optimal embedding and lag-space dimension of 2 since two time lags uniquely determine each value of x_k. The expected sensitivities to the two lags are $S(1) = 1.8959$ and $S(2) = 0.3$, respectively.

Figure 3a shows that the values of $\hat{S}(j)$ predicted by the neural network are in near perfect agreement with the expected values. In particular, $\hat{S}(j) < 2{\times}10^{-3}$ for $j > 2$. Error bars representing the standard deviation of the ten cases that were evaluated are not shown because they are negligibly small, typically about 1%. Figure 3b shows values of $F(j)$, defined as the fraction of neighbors that were false in dimension $j - 1$, plotted in this unconventional way so that the fraction reaches zero when the optimal embedding is obtained to simplify the comparison with $\hat{S}(j)$. Figure 3c shows the difference in calculated correlation dimension $\Delta D_2 = D_2(j) - D_2(j-1)$, which is expected to fall to zero once j exceeds the optimal embedding dimension, indicating that a plateau in the calculated D_2 has been reached. All three methods accurately predict an optimal embedding of 2.

Since the Hénon map is a relatively trivial example, the method was tested on several additional cases of increasing dimension and complexity. The first of these is the three-dimensional chaotic map from the preface of Ref. [18] whose form

$$x_k = x_{k-1}^2 - 0.2x_{k-1} - 0.9x_{k-2} + 0.6x_{k-3}, \tag{6}$$

gives the strange attractor shown in Figure 2b with an optimal embedding dimension of 3. The expected sensitivities are $S(1) = 1.1502$, $S(2) = 0.9$, and $S(3) = 0.6$. Figure 4 compares the three methods, and they show reasonable agreement, except that the plateau for the correlation dimension appears to be closer to 2 than to 3 as expected since the attractor has a dimension less than 2, and hence the overlaps are a set of measure zero.

A four-dimensional example in which the lag space is less than the optimal embedding dimension is the delayed Hénon map [19],

$$x_k = 1 - 1.6x_{k-1}^2 + 0.1x_{k-4}, \tag{7}$$

with an attractor shown in Figure 2c. The dynamics of the map only depend on the first and fourth time lags. The expected sensitivities are $S(1) = 1.9018$, $S(2) = S(3) = 0$, and $S(4) = 0.1$.

Figure 5 shows that only the neural network method identifies the gap in the time lags where the sensitivities that should be zero; $\hat{S}(2)$ and $\hat{S}(3)$, are an order of magnitude smaller than $\hat{S}(4)$. Unlike the saturation of the correlation dimension, false nearest neighbors accurately identifies 4 as the optimal embedding dimension. Another case (not shown) is a delayed Hénon map with an extremely long delay of 80, having expected sensitivities $S(1) = 1.9018$ and $S(80) = 0.1$. The sensitivities predicted by the trained network are $\hat{S}(1) = 1.6985$ and $\hat{S}(80) = 0.1035$, with all intermediate lags on the order of 10^{-3} or less. For this case, the neural network had 4 neurons, 80 dimensions, and a training error of 2.6212×10^{-3}, although more accurate sensitivities are likely with further training.

The final example is a four-dimensional variation of the Hénon map studied by Goutte [20] and given by

$$x_k = 1 - 1.4x_{k-2}^2 + 0.3x_{k-4}, \tag{8}$$

which, like the delayed Hénon map, has gaps in its lag space. Its strange attractor as shown in Figure 3d consists of two coexisting and non-interacting Hénon maps, one for odd k and the other for even k. The expected sensitivities are $S(2) = 1.8959$ and $S(4) = 0.3$, the same as for the simple Hénon map but with different lags.

Figure 6 shows that the neural network method works very well, while the other two methods predict an incorrect embedding of 3. Only the neural network method correctly identifies the gaps in the time lags. It is remarkable that with only four neurons, the neural network is able to accurately model two co-mingled two-dimensional nonlinear maps.

4 Data Requirements and Noise

In the real world, data records are often short and contaminated with noise. The performance of the various methods was tested using time series for the simple Hénon map of different lengths and with added noise. The performance was compared by calculating the normalized root mean square error for each method. For example, the error for the neural network method is given by

$$E = \sqrt{\frac{\sum_{j=1}^{d} (\hat{S}(j) - S(j))^2}{\sum_{j=1}^{d} S^2(j)}}, \tag{9}$$

and similarly for the other two methods. For false nearest neighbors, the expected values were determined from a calculation using 6,000 noise-free data points from the simple Hénon map. For the correlation dimension, the expected values were determined from an estimation using 10,000 data points from the simple Hénon map.

Figure 7 shows the performance of the three methods for varying amounts of data. The neural network method works almost perfectly even with as few as 32 data points, whereas the other methods seriously degrade when the number is less than several hundred.

In all the previous examples, the neural network had fixed values of n and d, chosen to be adequate for the cases studied. With experimental data, one would not typically know in advance how to choose d in particular. In such a case, one could train the network with increasing values of d, looking for a knee in a plot of the error e versus d, signifying that an adequate embedding had been achieved, much as one does for false nearest neighbors and for the correlation dimension. Figure 8 shows such a plot for a variant of the delayed Hénon map in Equation 7, but with a delay of 5,

$$x_k = 1 - 1.6x_{k-1}^2 + 0.1x_{k-5}, \tag{10}$$

This modification results in expected sensitivities $S(1) = 1.9018$ and $S(5) = 0.1$. Figure 8 shows that as d is increased with $n = 4$ and $c = 512$, the root mean square error e falls by a factor of 100 at $d = 5$ and remains at that level, signifying that the optimal embedding for next-step prediction was reached. The normalized root mean square error E in the sensitivities $S(j)$ increases slightly and then falls by a factor of 2 at $d = 5$. The knee would have been even more pronounced for a map with a larger value of $S(5)$, which in this case is only about 5% of $S(1)$.

To compare the methods in the presence of noise, Gaussian white noise of varying amounts was added to a time-series with $c = 512$ from the simple Hénon map using Equation 9 to compare the three methods, where the expected values were taken from a noiseless case with 512 data points from the simple Hénon map. The results in Figure 9 show that the neural network method

is considerably more robust to noise than are the other methods, but as the signal-to-noise ratio approaches unity, it too fails.

Since it is well known that colored noise can masquerade for low-dimensional chaos [21], the same experiment was performed with integrated white noise (also known as Brownian or $1/f^2$ noise) with the results shown in Figure 10. The superiority of the neural network method over the other methods is even more evident than for the case with white noise, probably because the noise is concentrated at low frequencies and is relatively small in the band of frequencies occupied by the signal. Presumably, this result would also hold for other forms of colored noise.

The typical error bars in Figures 9 and 10 at a signal-to-noise ratio of 7 dB indicate the standard deviation from the mean of ten different instances of the time series (different signal and different noise) for each of the indicated cases. It is not surprising that the neural network completely removes the noise and gives a low-dimensional attractor since it is entirely deterministic, but at the expense of some distortion of the signal. Any method that fits a deterministic model to data by its nature will be noise-free, and thus noise reduction is a byproduct of the method proposed here.

With real-world data, it is not usually clear what is signal and what is noise, and thus one can never be confident how much of the difference between the model and the data is really noise. For that purpose, a number of additional metrics have been proposed including the (ε, τ)-entropy [22, 23], which generalizes the Kolmogorov-Sinai entropy, the finite-size Lyapunov exponent [15, 16], which reduces errors due to low-level noise, and the scale-dependent Lyapunov exponent [24], which extends the finite-size Lyapunov exponent to a range of sizes and is especially useful for short time series. These metrics may sometimes be useful for avoiding errors in the embedding caused by noise and to optimize

the noise reduction, but they are not required for the systems studied in this paper since we have the luxury of knowing the correct embedding directly from the form of the maps that were used to test the method.

5 Conclusions

A proposed neural network method for determining the optimal embedding dimension for short-term predictability and lag space was tested for time series generated from various chaotic time-delayed maps and was found to work almost perfectly, even with a very small number of neurons. This method performs much better than false nearest neighbors and the plateau in the correlation dimension, neither of which are capable of determining gaps in the embedding dimension. Furthermore, the neural network method gives quantitative information about the relative importance of each time lag in next- step prediction, which the other methods do not. When the time series is short and/or contaminated by noise, the neural network method degrades more gracefully than the other methods.

We further remark that the method does not require that the model be a neural network or that it use a hyperbolic tangent basis function. Additionally, this method could use other universal approximators such as support vector machines [25], by perturbing the inputs to the network one at a time and measuring its affect on the output. If one has the luxury of having a physically based model for the system under study, it is almost always a good idea to use that model rather than a general one such as a neural network, but it must be a model of sufficiently high dimension with a corresponding large number of parameters.

It remains to be seen how well the method works for more complicated data that have been processed by an arbitrary transformation or those obtained from

sampling a continuous-time system where more complicated network architectures may be required. Even more interesting is to apply the method to time series records from experimental or observational data. These studies are beyond the scope of this paper and will be the subject of future publications.

References

[1] N.H. Packard, J.P. Crutchfield, J.D. Farmer, R.S. Shaw, Geometry from a time series, Phys. Rev. Lett. 45 (1980) 712–716.

[2] F. Takens, Detecting strange attractors in turbulence, in: D.A. Rand, L.S. Young (Eds.), Dynamical Systems and Turbulence, Springer, Berlin, 1981, pp. 366–381.

[3] T. Sauer, J. Yorke, M. Casdagli, Embedology, J. Stat. Phys. 65 (1991) 570–616.

[4] P. Grassberger, I. Procaccia, Measuring the strangeness of strange attractors, Phys. D. 9 (1983) 189–208.

[5] E. Allman, J. Rhodes, Mathematical Models in Biology: An Introduction, Cambridge, New York, 2004.

[6] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, Time-series Analysis: Forecasting and Control, third ed., Prentice-Hall, Englewood Cliffs, NJ, 1994.

[7] H. Abarbanel, Analysis of Observed Chaotic Data, Springer, New York, 1996.

[8] C.J. Cellucci, A.M. Albano, P.E. Rapp, Comparative study of embedding methods, Phys. Rev. E. 67 (2003) 066210.

[9] M. Kennel, R. Brown, H. Abarbanel, Determining embedding dimension for phase-space reconstruction using a geometrical construction, Phys. Rev. A. 45 (1992) 3403–3411.

[10] S. Troncia, M. Gionab, R. Barata, Reconstruction of chaotic time series by neural models: a case study, Neurocomputing 55 (2003) 3–4.

[11] J. Principe, J. Kuo, Dynamic modelling of chaotic time series with neural networks, in: G. Tesauro, D. Touretzky, T. Leen (Eds.) Advances in Neural Information Processing Systems, Denver, 1994, pp. 311.

[12] G. Zhang, B. Patuwo, M. Hu, Forecasting with artificial neural networks: The state of the art, Int. J. Forecasting 14 (1998) 35–62.

[13] J. Principe, A. Rathie, J. Kuo, Prediction of chaotic time series with neural networks and the issue of dynamic modeling, Int. J. Bifurcation Chaos 2 (1992) 989-996.

[14] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks 3 (1990) 535–549.

[15] E. Aurell, G. Boffetta, A. Crisanti, G. Paladin, A. Vulpiani, Predictability in the large: an extension of the concept of Lyapunov exponent, J. Phys. A: Math. Gen. 30 (1997) 1–26.

[16] C. Ziehmann, L. A. Smith, J. Kurths, Localized Lyapunov exponents and the prediction of predictability, Phys. Lett. A. 271 (2000) 237–251.

[17] M. Hénon, A two-dimensional mapping with a strange attractor, Comm. Math. Phys. 50 (1976) 69–77.

[18] J.C. Sprott, Chaos and Time-Series Analysis, Oxford, New York, 2003.

[19] J.C. Sprott, High-dimensional dynamics in the delayed Hénon map, Electron. J. Theor. Phys. 3 (2006) 19–35.

[20] C. Goutte, Lag space estimation in time series modelling, in: B. Werner (Ed.) IEEE International Conference on Acoustics, Speech, and Signal Processing, Munich, 1997, pp. 3313.

[21] A.R. Osborne, A. Provenzale, Finite correlation dimension for stochastic systems with power-law spectra, Phys. D. 35 (1989) 357–381.

[22] P. Gaspard, X.-J. Wang, Noise, chaos, and (ε, τ)-entropy per unit time, Phys. Rep. 235 (1993) 291–343.

[23] M. Cencini, M. Falcioni, E. Olbrich, H. Kantz, A. Vulpiani, Chaos or noise: Difficulties of a distinction, Phys. Rev. E. 62 (2000) 427–437.

[24] J.B. Gao, J. Hu, W.-W. Tung, Y.H. Cao, Distinguishing chaos from noise by scale-dependent Lyapunov exponent, Phys. Rev. E. 74 (2006) 066204.

[25] B. Hammer, K. Gersmann, A note on the universal approximation capability of support vector machines, Neural Proc. Lett. 17 (2001) 43–53.

Figure Captions

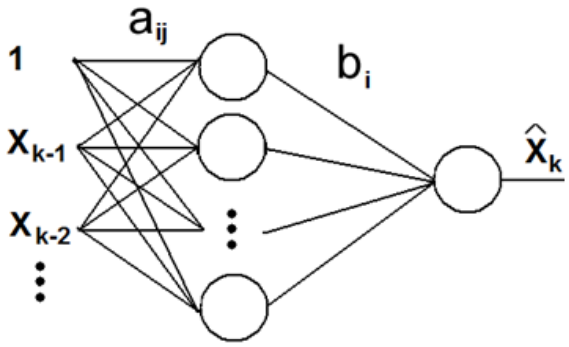

Figure 1: Single layer, feed-forward neural network

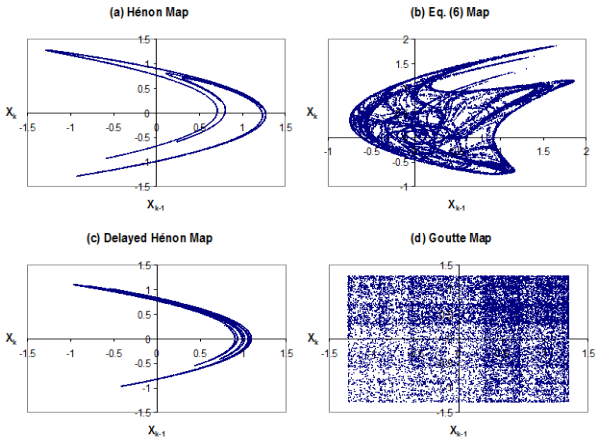

Figure 2: Strange attractors of maps studied

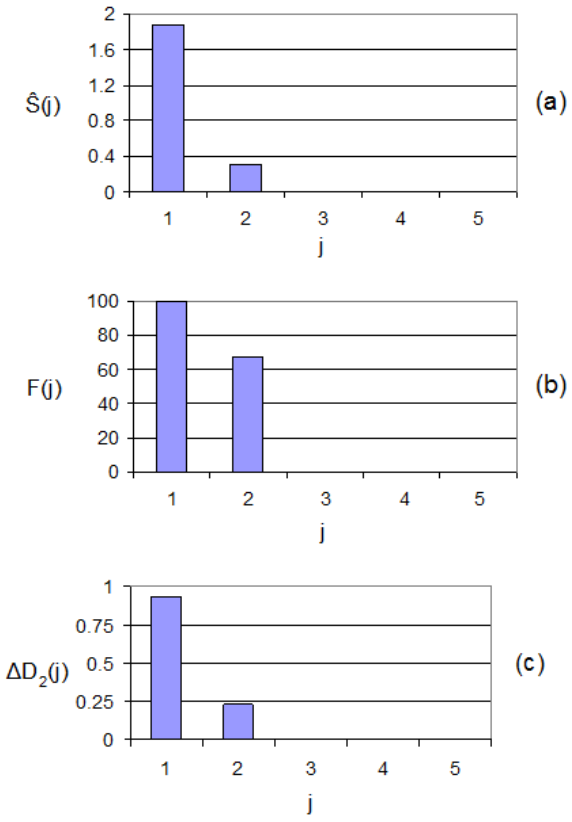

Figure 3: Embedding calculations for simple Hénon map using (a) Neural network sensitivities (b) False nearest neighbors (c) Correlation dimension

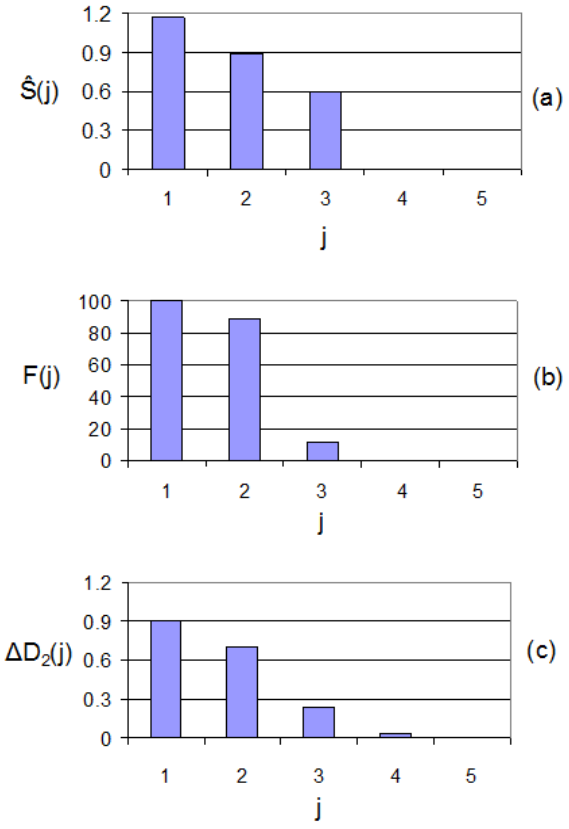

Figure 4: Embedding calculations for Eq. (6) using (a) Neural network sensitivities (b) False nearest neighbors (c) Correlation dimension

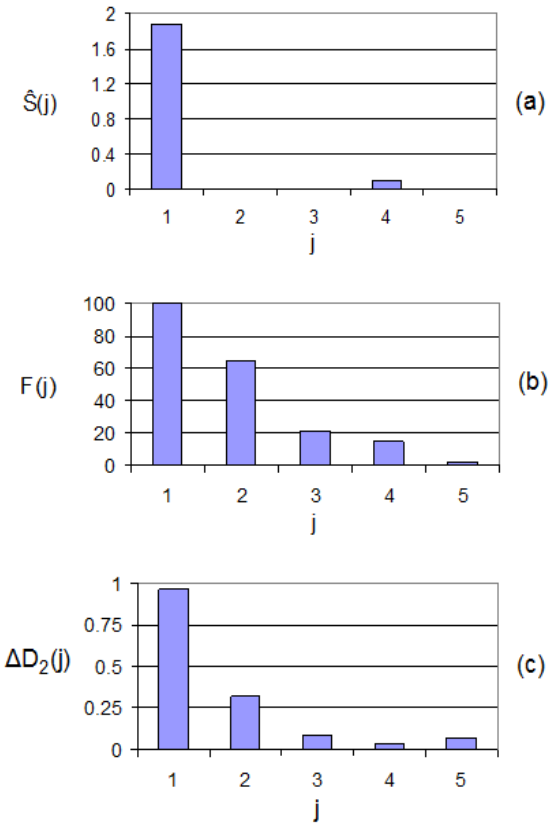

Figure 5: Embedding calculations for delayed Hénon map using (a) Neural network sensitivities (b) False nearest neighbors (c) Correlation dimension

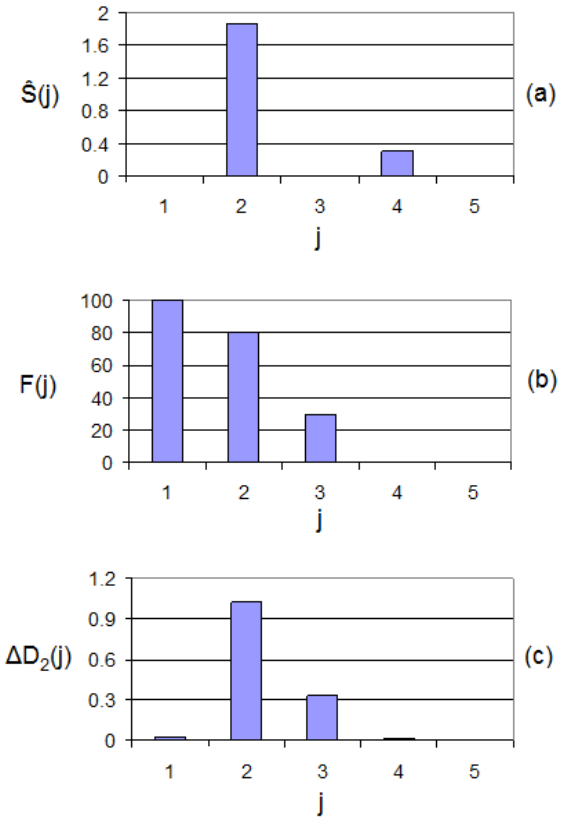

Figure 6: Embedding calculations for Goutte map using (a) Neural network sensitivities (b) False nearest neighbors (c) Correlation dimension

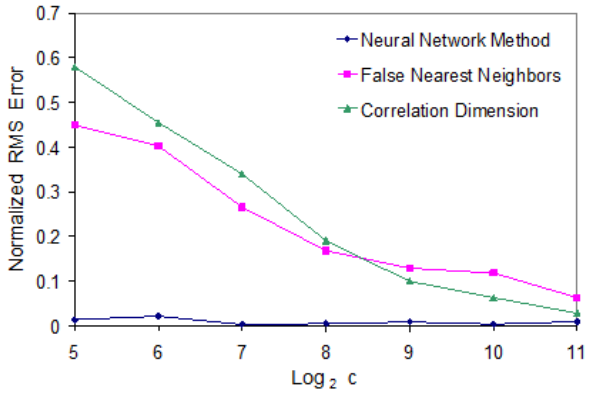

Figure 7: Comparison of three methods for calculating embedding dimension versus length of the time series c for the Hénon map

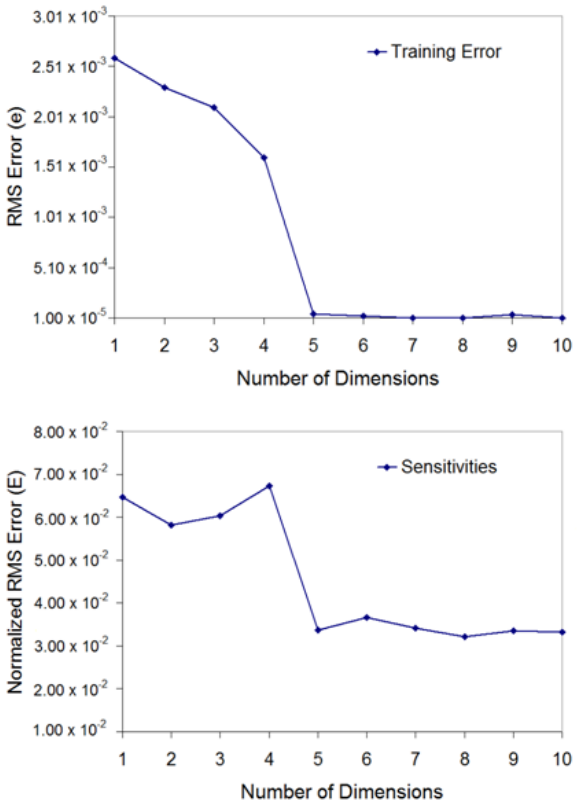

Figure 8: Change in neural network training error and sensitivities while varying dimensions for the delayed Hénon map

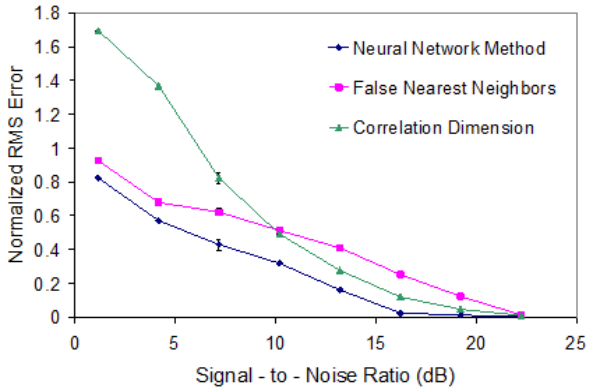

Figure 9: Comparison of three methods for calculating embedding dimension in the presence of varying degrees of White noise for the Hénon map

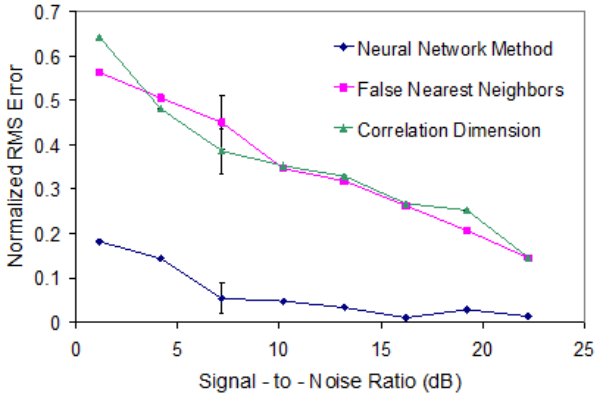

Figure 10: Comparison of three methods for calculating embedding dimension in the presence of varying degrees of Brownian noise for the Hénon map