2005 International Symposium on Nonlinear Theory and its Applications (NOLTA2005)
Bruges, Belgium, October 18-21, 2005

Estimation of Embedding Dimension Using Self-Organizing Map

Haruna MATSUSHITA and Yoshifumi NISHIO

Department of Electrical and Electronics Engineering
Tokushima University
2-1 Minami-Josanjima, Tokushima 770-8506, Japan
Phone:+81-88-656-7470, Fax:+81-88-656-7471
Email: {haruna, nishio}@ee.tokushima-u.ac.jp

Abstract—In order to analyze the nonlinear time series, the embedding is very important. In the embedding process, We must determine the appropriate embedding dimension to reconstruct with time-delay. In this study, we propose a method to estimate the embedding dimension using Self-Organizing Map (SOM). We can obtain the map reflecting the distribution state of input data using SOM. We carry out simulations for the time series obtained from the Henon map, the Ikeda map, and the Lorenz equations, and confirm that our method would be useful to estimate the appropriate embedding dimensions.

1. Introduction

In the analysis of the time series come from nonlinear dynamical systems, the embedding is very important [1]. In the embedding process, the embedding dimension k must be properly determined. If the current embedding dimension k is too small, points which are really far apart will be brought artificially close together. Conversely, if the embedding dimension is too large, the structure of the reconstructed attractor becomes coarse, consequently, when we analyze the finite observed time series data, there is a sufficient possibility that the shortage of data will bring about the false analysis result. In addition, the calculation amount of analysis increase pro rata the embedding dimension, so, determining the appropriate embedding dimension is very important.

A common procedure for determining the embedding dimension is the False Nearest Neighbors method (FNN method) [2]. However, the calculation amount is quite large for this method. In this study, we propose a method to estimate the embedding dimension using Self-Organizing Map (SOM) [3]. SOM is unsupervised neural network introduced by Kohonen in 1982. Self-Organization is to change an internal structure to adjust to the signal from the outside. SOM is a model simplifying self-organization process of brain. SOM obtains a statistical feature of input data. We can obtain the map reflecting the distribution state of input data using SOM. The motivations of our method are as follows: 1) SOM could grasp the statistical characteristics of the distribution of the points in the reconstructed attractors and 2) the characteristics would change dramatically until the embedding dimensions reaches to the appropriate value and would not change beyond the value. We carry out computer simulations for the time series obtained from the Henon map, the Ikeda map, and the Lorenz equations with different delay time, and confirm that our method would be useful to estimate the appropriate values of the embedding dimensions.

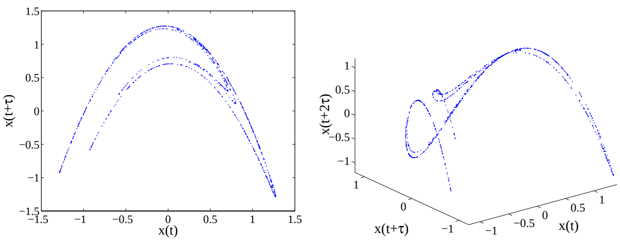
2. Embedding

In order to analyze the nonlinear time series, reconstruction of attractors must be done first. Suppose the d-dimensional state vector $x(t)$ evolves according to an unknown but continuous and deterministic dynamics. In particular, we construct the state vector $\boldsymbol{y}(t)$ with delay coordinates in the k-dimensional reconstructed space as follows;

$$\boldsymbol{y}(t) = [x(t), x(t+\tau), \cdots, x(t+(k-1)\tau)], \qquad (1)$$

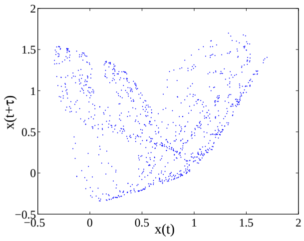
where k is the embedding dimension and τ is the delay time. The trajectory in the reconstructed state space is defined as a sequence of k-dimensional vectors.

Figures 1-3 show the reconstructed attractors for the time series obtained from the Henon map, the Ikeda map, and the Lorenz equations with different delay times, respectively, where, the total number of the data points is $N = 1000$.

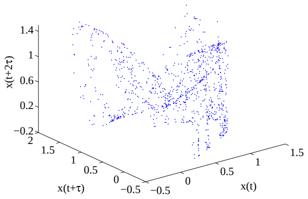


(a) $k = 2$. (b) $k = 3$.

Figure 1: Reconstructed attractor for Henon map.

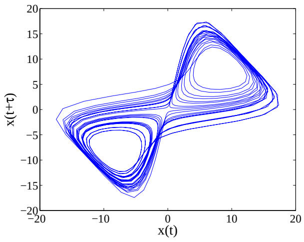


(a) $k = 2$.

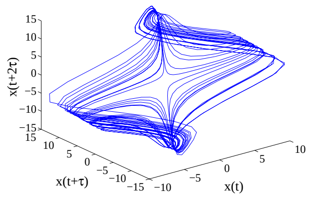


(b) $k = 3$.

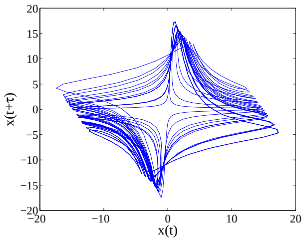
Figure 2: Reconstructed attractor for Ikeda map.



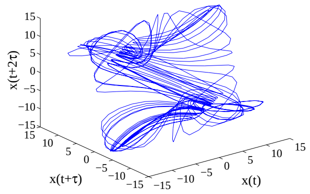
(a) $k = 2$, $\tau = 5$.



(b) $k = 3$, $\tau = 5$.



(c) $k = 2$, $\tau = 10$.



(d) $k = 3$, $\tau = 10$.

Figure 3: Reconstructed attractor for Lorenz equations.

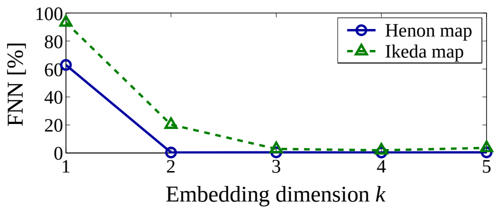
3. False Nearest Neighbor method

The FNN method is that if the current embedding dimension k is sufficient to resolve the dynamics, points which were close each other in the k-dimensional reconstructed attractor should remain close in the $(k+1)$-dimensional reconstructed attractor. Conversely, if the embedding dimension is too small, points which are really far apart will be brought artificially close together.

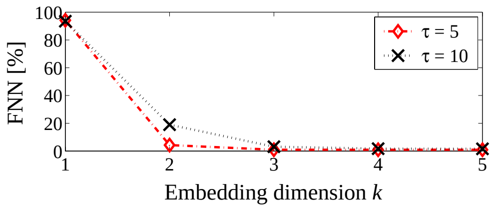
The FNN algorithm is explained as follows. Take one point in the k-dimensional reconstructed attractor, find its nearest neighbor in the attractor, and then calculate the distance between them. Re-calculate the distance between the same points, but in the $(k+1)$-dimensional reconstructed attractor. If the recalculated distance is more than a certain fixed multiple of the original distance, these two points are said to be "false nearest neighbors (FNN)." The fixed multiple is chosen as from 2 to 15. Repeat the process for all of the points in the k-dimensional reconstructed attractor. If the number of the false nearest neighbors is not smaller than a sufficiently small value, the embedding dimension is defined to be too small.

Figure 4 shows the simulation results of FNN method for the Henon map, the Ikeda map, and the Lorenz equations with $\tau = 5$ and 10. The horizontal axis is the embedding dimension and the vertical axis is the number of the false nearest neighbors (percentage).

We can see from the figure that $k = 2$ is enough for the Henon map and the Lorenz equations with $\tau = 5$. While the Ikeda map and the Lorenz equations with $\tau = 10$ have some amount of false nearest neighbors for $k = 2$ and hence $k = 2$ is not sufficient to embed the information of these time series. For $k = 3$, the numbers of the false nearest neighbors becomes almost zero for all of the time series data.



(a) Henon map and Ikeda map.



(b) Lorenz equations with different delay time.

Figure 4: Simulation results of FNN method.

4. SOM method

4.1. Algorithm of SOM method

We explain the learning algorithm of SOM and the method of determining the embedding dimension using SOM. m neurons are connected on a one-dimensional line like a rope (where the first neuron and m_{th} neuron are not connected). The input vectors $\boldsymbol{y}(t)$ are the points in the k-dimensional reconstructed attractor.

[PHASE 1]
(SOM1) The initial values of all the weight vectors $\boldsymbol{w}$ are given over the input space at random.

[PHASE 2]
(SOM2) An input vector $\boldsymbol{y}(t)$ is inputted to all the neurons at the same time in parallel.

(SOM3) We find the winner neuron $c(t)$ by calculating the distances between the input vector $\boldsymbol{y}(t)$ and the weight vector $\boldsymbol{w}_i = (w_{i1}, w_{i2}, \cdots, w_{ik})$ $(i = 1, 2, \cdots, m)$ of neuron i, according to;

$$c(t) = \arg\min_i \{\|\boldsymbol{w}_i - \boldsymbol{y}(t)\|\}. \qquad (2)$$

In other words, the winner neuron $c(t)$ is the neuron with the weight vector nearest to the input vector $\boldsymbol{y}(t)$. In this study, Euclidean distance is used for (2).
(SOM4) The weight vector of the neurons are updated as;

$$\boldsymbol{w}_i(s+1) = \boldsymbol{w}_i(s) + h_{c(t),i}(s)(\boldsymbol{y}(t) - \boldsymbol{w}_i(s)), \qquad (3)$$

where s is the learning step. $h_{c(t),i}(s)$ is called the neighborhood function and it is described as follows;

$$h_{c(t),i}(s) = \alpha(s)\exp\left(-\frac{\|\boldsymbol{r}_i - \boldsymbol{r}_{c(t)}\|^2}{2\sigma^2(s)}\right), \qquad (4)$$

where $\alpha(s)$ is the learning rate, $\boldsymbol{r}_i$ and $\boldsymbol{r}_{c(t)}$ are the vectorial locations on the display grid, and $\sigma(s)$ corresponds to the widths of the neighborhood functions. $\alpha(s)$ and $\sigma(s)$ decrease with time according to the following equations;

$$\begin{aligned} \alpha(s) &= \alpha(0)\,(1 - s/T)\,, \\ \sigma(s) &= \sigma(0)\,(1 - s/T)\,, \end{aligned} \qquad (5)$$

where T is the maximum number of the learning.
(SOM5) The steps from (SOM2) to (SOM4) are repeated for all the input data.

[PHASE 3]
(SOM6) After learning, we calculate the distance L between the weight vectors of the adjacent two neurons.

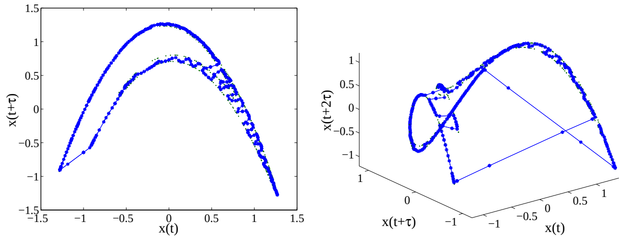
$$L = \frac{\|\boldsymbol{w}_j - \boldsymbol{w}_{j+1}\|}{\sqrt{k}}, \quad (j = 1, 2, \cdots, m-1). \qquad (6)$$

4.2. Simulation Results

We simulated the method of determining the embedding dimension using SOM. Input data are $N = 1000$ points obtained from the Henon map, the Ikeda map, and the Lorenz attractors with different τ, which are the same as the data in the FNN method. The simulation was performed as changing the embedding dimension k from 1 to 5. Each SOM has $m = 300$ neurons. The parameters of the learning are chosen as follows;

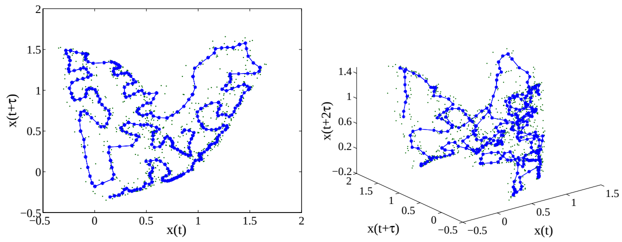
$$\alpha(0) = 0.95, \quad \sigma(0) = 9.$$

We repeat [PHASE 2] 20 times (namely the maximum number of the learning is $T = 20000$). The results of the learning for the cases of $k = 2$ and $k = 3$ are shown in Figs. 5-7. We can see that the SOMs can obtain statistical features of the reconstructed attractors.



(a) $k = 2$. (b) $k = 3$.

Figure 5: Learning of SOM for Henon map.



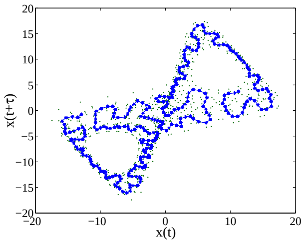
(a) $k = 2$. (b) $k = 3$.

Figure 6: Learning of SOM for Ikeda map.

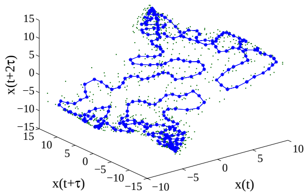
After [PHASE 2], we calculate the distance L according to [PHASE 3]. The probability density functions of the distances L for all of the cases are shown in Fig. 8. We can see that there are obvious differences between the graphs of $k = 1$ and $k = 2$, however, are not big differences between $k = 2$, 3, 4 and 5. This is attributed to the fact that $k = 1$ is too small to embed the information of the time series for all of the data and that $k = 2$ is enough or somewhat close to appropriate value. These results agree well the results obtained by using the FNN method in Fig. 4.

Let us observe the graphs more carefully paying attentions on the graphs corresponding to $k = 2$. The results for the Ikeda map are shown in Fig. 8(b). Among the graphs corresponding to $k = 2$, 3, 4, and 5, only the graph of $k = 2$ is a little bit apart from the others. This result suggests that some information would be submerged in the 2-dimensional reconstructed attractor. Similar feature can be seen in Fig. 8(d). Namely, for the data obtained from the Lorenz attractors with $\tau = 10$, embedding to the 2-dimensional state space submerges some information of the time series.

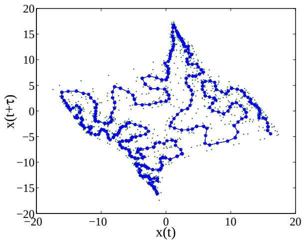
As a result from these figures, we could estimate the appropriate values of the embedding dimensions as $k = 2$, $k = 3$, $k = 2$, and $k = 3$ for the Henon map, the Ikeda map, the Lorenz attractors with $\tau = 5$, and the Lorenz attractors with $\tau = 10$, respectively.



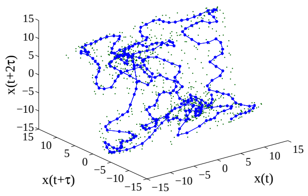
(a) $k = 2$, $\tau = 5$.



(b) $k = 3$, $\tau = 5$.



(c) $k = 2$, $\tau = 10$.



(d) $k = 3$, $\tau = 10$.

Figure 7: Learning of SOM for Lorenz attractors.

5. Conclusion

In this study, we have proposed the method of determining the embedding dimension using Self-Organizing Map. By using this method, we can determine the embedding dimension, and obtain the transit of the reconstructed state. In the future, we confirm the effectiveness of this method by applying to the higher dimensional and more complex chaos.

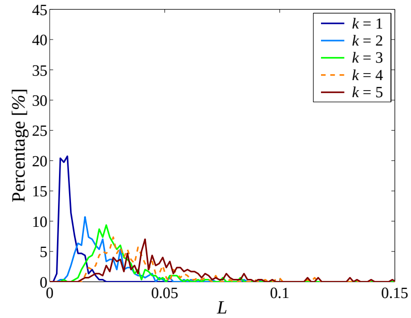
References

[1] N. H. Packard, J. P. Crutchfield, J. D. Farmar, and R. S. Shaw, "Geometry from a Time Series," *Physical Review Letters*, vol.45, No.9, pp.712-716, 1980.

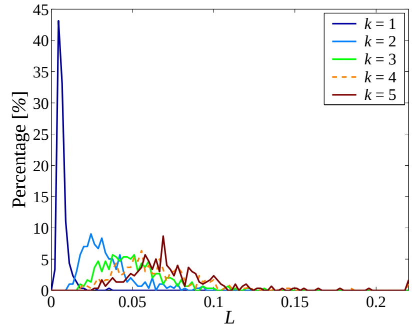
[2] Matthew B. Kennel and Henry D. I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Physical Review A*, vol.45, No.6, pp.3403-3411, 1992.

[3] T. Kohonen, *Self-Organizing Maps*, Berlin, Springer, vol. 30, 1995.

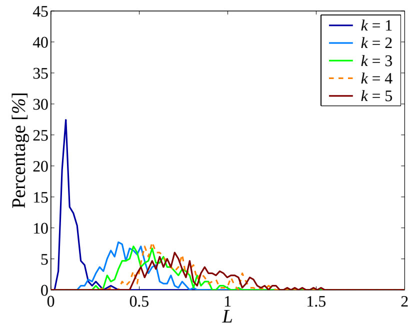
[4] Matthew B. Kennel and Henry D. I. Abarbanel, "False Neighbors and False Stands: A Reliable Minimum Embedding Dimension Algorithm," *Physical Review E*, vol.66, 2002.



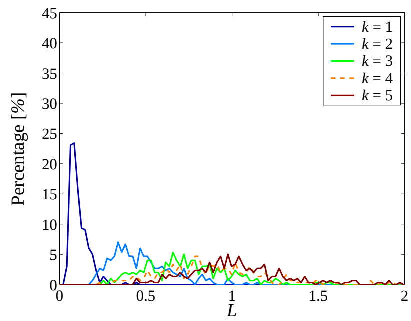
(a) Henon map.



(b) Ikeda map.



(c) Lorenz attractors ($\tau = 5$).



(d) Lorenz attractors ($\tau = 10$).

Figure 8: Probability density functions of distances L between the weight vectors of the adjacent two neurons.