Reconstructing state spaces from multivariate data using variable delays

Yoshito Hirata,[1,*] Hideyuki Suzuki,[1,2] and Kazuyuki Aihara[1,2,3]
[1]*Department of Mathematical Informatics, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan*
[2]*Institute of Industrial Science, The University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505, Japan*
[3]*Aihara Complexity Modelling Project, ERATO, JST, 45-18 Oyama, Shibuya-ku, Tokyo 151-0065, Japan*
(Received 31 January 2006; revised manuscript received 12 May 2006; published 2 August 2006)

We study two methods for constructing a nonuniform embedding for multivariate data. A nonuniform embedding is a state space reconstruction which is more flexible than the common delay coordinates with fixed delays since it contains variable delays. Using these methods, we can extract causal relationships among many variables in a more suitable way. We demonstrate that the proposed methods can give more precise predictions and simpler models than some previous methods.

DOI: 10.1103/PhysRevE.74.026202 PACS number(s): 05.45.Tp, 05.45.Jn

Multivariate time series are ubiquitous in the fields of science, technology, and economics. Typical examples include multichannel recordings of the brain, weather forecasting, genetic networks, food chains, robotics, and prices of stocks and economic indexes. In these areas of research, a major interest is to infer some relations among many variables.

In a nonlinear time series analysis for a scalar time series, the story often starts from an embedding: the reconstruction of a state space from observed data. The history of embedding dates back to 1980, when Packard *et al.* [1] proposed the method of "delay coordinates" and Takens [2] gave it theoretical support. Given a scalar time series $\{x(t)\}_{t=1}^{N}$, the *delay coordinates* are defined as $(x(t), x(t-\tau), \ldots, x(t-(d-1)\tau))$, where τ is called the *lag*, and d the *embedding dimension*. The delay coordinates will be later called a *uniform embedding* for the sake of comparison. According to Takens's theorem [2], if d is bigger than twice the dimension of the original dynamical system, the delay coordinates produce an embedding, a faithful representation of the original state. Later Sauer *et al.* [3] showed that the method of delays preserves the box-counting dimension. Stark and co-workers extended Takens's theorem toward forced and stochastic systems [4–6]. In practice Cellucci *et al.* [7] recently argued that τ and d are best identified from a scalar time series using the first minimum of the mutual information [8] and false nearest neighbors [9], respectively.

Although it is effective when a time series has a single dominant frequency, a uniform embedding does not work well when a time series has multiple strong periodicities with different time scales [10]. A uniform embedding fails because a lag cannot be optimal for both short and long time scales. A time series that possesses multiple strong periodicities is, for example, that of wind velocity. A solution for multiple periodicities is "nonuniform embeddings" [10]. By choosing a *lag vector* $(\tau_1, \tau_2, \ldots, \tau_k)$, we define a *nonuniform embedding* as $(x(t-\tau_1), x(t-\tau_2), \ldots, x(t-\tau_k))$, where k is the number of lags and $\tau_i < \tau_j$ if $i<j$. A nonuniform embedding can deal with several time scales simultaneously because short lags describe short-time-scale dynamics while long lags capture the dynamics of long time scales. Some methods for constructing a nonuniform embedding for a scalar time series have been proposed [10–14]. Nonuniform embeddings are useful for better description of phenomena since short and long periodicities can be dealt with together [10,13] and more precise prediction [12] is possible.

In the case of multivariate time series, it is often the case that a time series contains multiple periodicities. There are also some proposed methods for finding uniform embeddings [15,16]. As for nonuniform embeddings, Garcia and Almeida [17] proposed an algorithm that further extends their extension [14] of false nearest neighbors [9], a standard practical method for finding a uniform embedding for a scalar time series.

In this paper, we examine two effective methods for constructing a nonuniform embedding given a multivariate time series. One is an extension of the method of Judd and Mees [10], the other is a different technique that we call cross validation. These will help provide not only better predictions but also more convenient descriptions.

Before we start describing its extension, we briefly summarize the method of Judd and Mees [10], which finds a nonuniform embedding from a scalar time series.

Let N be the number of observations in a time series. Suppose that there is a scalar time series $\{x(t)\}_{t=1}^{N}$ given. Then a standard linear autoregressive model has the following form:

$$x(t) = a_0 + \sum_{\delta=1}^{w} a_\delta x(t-\delta) + \varepsilon_t. \tag{1}$$

This can be considered as a linear model for a uniform embedding of dimension w and a unit lag.

We may want to choose the lag vector optimally and remove the terms that do not contribute significantly. Assume that we have a lag vector $(\tau_1, \tau_2, \ldots, \tau_k)$ where $\tau_k \leqslant w$. Then a reduced autoregressive linear model has the following form:

$$x(t) = a_0 + \sum_{j=1}^{k} a_j x(t-\tau_j) + \varepsilon_t. \tag{2}$$

*Electronic address: yoshito@sat.t.u-tokyo.ac.jp

For selecting an optimal set of lags, Judd and Mees [10] used the minimum description length [18]. The description length is a tool for changing the approximation errors and the size of the model into a single criterion so that we can find an optimal tradeoff between the approximation errors and the model size. First we regard terms $\{x(t-\delta):\delta=1,\ldots,w\}$ as candidate basis functions for linear approximation. Then we minimize the description length and declare a set of basis functions giving the minimum as the optimal set.

In a similar way to Judd and Mees [10], we define a method for finding a nonuniform embedding from a multivariate time series. First we define a multivariate reduced autoregressive linear model. Suppose hereafter that a multivariate time series $\{x(t)\}_{t=1}^{N}$ is given, where $x(t)=(x_1(t),\ldots,x_D(t))$ is defined to be a vector of all D observables at time t. Then a *multivariate reduced autoregressive linear model* for the ith observable has the following form:

$$x_i(t) = a_0 + \sum_{j=1}^{k} a_j x_{i_j}(t-\tau_j) + \varepsilon_t, \tag{3}$$

where k is the number of delays and i_j is the index of the observable for the jth delay; thus it takes an integer value between 1 and D. In this model, the set of candidate basis functions is $\{x_i(t-\delta):i=1,\ldots,D;\delta=1,\ldots,w\}$ and the set of chosen basis functions is $\{x_{i_j}(t-\tau_j):j=1,\ldots,k\}$. To obtain a reduced model, we need to find a set of basis functions that best describes the given time series. Instead of the description length, here we use the normalized maximum likelihood [19], a variant of the description length, for the information criterion. The normalized maximum likelihood is defined as follows. Suppose that there are a number K of candidate basis functions $\{f_j:j\in F\}$, where $F=\{1,2,\ldots,K\}$ is the set of indices for all the candidates. Let v_t be a vector $(x_1(t-1),x_1(t-2),\ldots,x_1(t-w),x_2(t-1),\ldots,x_D(t-w))$. For a set of indices $B=\{j_1,\ldots,j_k\}\subset F$, define

$$V_B = \begin{pmatrix} f_{j_1}(v_{w+1}) & f_{j_2}(v_{w+1}) & \cdots & f_{j_k}(v_{w+1}) \\ \vdots & & & \vdots \\ f_{j_1}(v_t) & f_{j_2}(v_t) & \cdots & f_{j_k}(v_t) \\ \vdots & & & \vdots \\ f_{j_1}(v_N) & f_{j_2}(v_N) & \cdots & f_{j_k}(v_N) \end{pmatrix}, \tag{4}$$

$$\xi = (x_i(w+1),\ldots,x_i(t),\ldots,x_i(N))^T, \tag{5}$$

$$\lambda_B = (a_1,a_2,\ldots,a_k)^T, \tag{6}$$

where the superscript T indicates transposition. Then the prediction error e_B can be written as $e_B=\xi-V_B\lambda_B$. By minimizing the squares $e_B^T e_B$ of the prediction errors over λ_B, we obtain the estimate $\hat{\lambda}_B$ of λ_B as

$$\hat{\lambda}_B = (V_B^T V_B)^{-1} V_B^T \xi. \tag{7}$$

Then the mean square $\hat{\tau}_B$ of the errors is

$$\hat{\tau}_B = (\xi - V_B\hat{\lambda}_B)^T(\xi - V_B\hat{\lambda}_B)/\tilde{N}, \tag{8}$$

where $\tilde{N}=N-w$. We also define

$$\hat{R}_B = (V_B\hat{\lambda}_B)^T V_B\hat{\lambda}_B/\tilde{N}. \tag{9}$$

Then the normalized maximum likelihood ($\mathcal{L}$) [19] for k basis functions $\{f_j:j\in B\}$ is defined as

$$\mathcal{L}(B) = \frac{\tilde{N}-w}{2}\ln\hat{\tau}_B + \frac{w}{2}\ln\hat{R}_B - \ln\Gamma\left(\frac{\tilde{N}-w}{2}\right) - \ln\Gamma\left(\frac{w}{2}\right) - \ln w,$$

where Γ is the Gamma function. Nakamura *et al.* [20] concluded that the normalized maximum likelihood is most likely to choose the correct model among other model selection criteria.

Using the normalized maximum likelihood and the algorithm of Judd and Mees [21], we define an algorithm for selecting an optimal set of delays as follows. Let V be a matrix each row of which is a set of all candidate basis functions, or delays, all at the same instant t, namely,

$$V = V_F = \begin{pmatrix} f_1(v_{w+1}) & f_2(v_{w+1}) & \cdots & f_K(v_{w+1}) \\ \vdots & & & \vdots \\ f_1(v_t) & f_2(v_t) & \cdots & f_K(v_t) \\ \vdots & & & \vdots \\ f_1(v_N) & f_2(v_N) & \cdots & f_K(v_N) \end{pmatrix}. \tag{10}$$

(1) Normalize V so that each column has unit length.

(2) Let B and B' be empty sets.

(3) Let the prediction error $e_{B'}=\xi$ if B' is empty and $e_{B'}=\xi-V_{B'}\hat{\lambda}_{B'}$ otherwise. The best estimate $\hat{\lambda}_{B'}$ for a set B' of basis functions can be calculated by Eq. (7). Then each component of $\mu=V^T e_{B'}$ shows how closely the corresponding basis function matches the residual $e_{B'}$.

(4) Find the candidate basis function f_p that matches best. Let $p\in F$ be the index for the largest component in μ. Set $B'\leftarrow B'\cup\{p\}$.

(5) Find the basis function f_q that contributes least. Let $q\in B'$ be the index whose element of $\hat{\lambda}_{B'}$ is the smallest absolute value. If $p\neq q$, then $B'\leftarrow B'\backslash\{q\}$ and go to step 3.

(6) If B is empty or $\mathcal{L}(B')<\mathcal{L}(B)$, then set $B\leftarrow B'$ and go to step 3.

(7) Declare that B is the best combination of basis functions.

When we have many candidate delays, we may not be able to handle them at once. In such a case, we change the above algorithm in the following way. Divide F into small groups F_l $(l=1,2,\ldots,L)$. For example, in the following numerical simulations, we set F_l as a set of indices for basis functions $\{x_l(t-1),x_l(t-2),\ldots,x_l(t-w)\}$.

(1) Let B be an empty set.

(2) For $l=1$ to L, apply the above algorithm using $B\cup F_l$ as the set of indices for candidate basis functions and replace B with the chosen basis functions.

(3) Declare that B is the best combination of basis functions.

TABLE I. Chosen delays for coupled Rössler systems.

Method	Chosen delays
Boccaletti *et al.*	$x_1(t-1)$, $x_1(t-32)$, $x_1(t-63)$, $x_2(t-1)$, $x_2(t-31)$, $x_2(t-61)$
Extension of Judd and Mees	$x_1(t-1)$, $x_1(t-12)$, $x_1(t-22)$, $x_1(t-50)$, $x_2(t-1)$, $x_2(t-8)$, $x_2(t-22)$, $x_2(t-32)$, $x_2(t-50)$
Cross validation	$x_1(t-1)$, $x_1(t-3)$, $x_1(t-9)$
Garcia and Almeida	$x_1(t-1)$, $x_1(t-2)$, $x_1(t-3)$, $x_1(t-4)$, $x_1(t-5)$, $x_1(t-6)$, $x_1(t-7)$, $x_1(t-8)$, $x_1(t-12)$, $x_2(t-1)$, $x_2(t-2)$, $x_2(t-3)$, $x_2(t-4)$, $x_2(t-5)$, $x_2(t-6)$, $x_2(t-7)$

The second method we propose is cross validation. Given a multivariate time series $\{(x_1(t),x_2(t),\ldots,x_D(t))\}_{t=1}^{N}$, we randomly select the number M_1 of points for modeling, and the number M_2 of points for evaluation from the remaining points. For a given nonuniform embedding defined by a set of delays $G=(x_{i_1}(t-\tau_1),x_{i_2}(t-\tau_2),\ldots,x_{i_k}(t-\tau_k))$, we can fit a radial basis function model with the points for modeling using the algorithm of Judd and Mees [21] with the normalized maximum likelihood as the information criterion, and evaluate the prediction error $E(G)$ using the points for evaluation. Here the centers of the radial basis functions are chosen as prescribed in Ref. [21], namely, from the points for modeling, embedded in that space, added to a Gaussian distribution of mean 0 and standard deviation 0.3 times the standard deviation of the given data. We use the root mean square error for the prediction error. Therefore, we minimize the prediction error over nonuniform embeddings to find the best nonuniform embedding. We may minimize $E(G)$ using the genetic algorithm [22]. However, we found that the following deterministic algorithm relatively works well.

(1) Let H be the set of all possible delays.

(2) Let G be an empty set.

(3) Find $h \in H \backslash G$ that minimizes $E(G \cup \{h\})$.

(4) If $E(G \cup \{h\}) < E(G)$, then $G \leftarrow G \cup \{h\}$ and go to step 3.

(5) Declare that G is a set of delays that yields a good nonuniform embedding.

The possible delays may be decided from our belief about how far past observables can influence the future. In this paper, we choose the maximum delay w and prepare all the possible delays for each coordinate, i.e., the set $\{x_i(t-\delta): i=1,2,\ldots,D; \delta=1,2,\ldots,w\}$. In practice, by testing a case with a longer maximum delay, we may be able to ensure that the chosen maximum delay is large enough.

We call this algorithm the greedy algorithm. In this paper, we use $M_1=1000$ and $M_2=1000$.

We applied the above two algorithms to artificial data. For the comparisons of their performances, we have three alternative methods: those of Boccaletti *et al.* [16], Cao, Mees, and Judd [15], and Garcia and Almeida [17]. While the proposed methods choose a set of delays for each observable, the method of Boccaletti *et al.* [16] reconstructs a vector of delays that corresponds to each original state of the entire system with uniform delays. The method of Cao *et al.* [15] builds single states for each observable with uniform delays. The method of Garcia and Almeida [17] finds single states for the entire system with nonuniform delays. However, we only use the methods of Boccaletti *et al.* [16] and Garcia and Almeida [17] since the method of Cao, Mees, and Judd [15] soon suffers from combinatorial explosions.

TABLE II. Root mean square error (RMSE) for coupled Rössler systems.

Method	RMSE
Boccaletti *et al.*	0.1073 ± 0.0050
Extension of Judd and Mees	0.0451 ± 0.0110
Cross validation	0.0018 ± 0.0022
Garcia and Almeida	0.0377 ± 0.0196

We applied these methods to data of coupled Rössler systems, which were used by Boccaletti *et al.* [16]. We used the coupling strength $\epsilon=0.05$. We observed x_1 and x_2 every 0.05 unit time to obtain a time series $\{(x_1(t),x_2(t))\}_{t=1}^{10\,000}$ of length 10 000. Then we made a model that predicts $x_1(t)$ from the two observables up to time $(t-1)$. We used the first 9500 points for modeling and the remaining 500 points for calculating the root mean square error. For the two proposed methods and that of Garcia and Almeida [17], we considered maximum delays of 50 for each observable.

The results are summarized in Tables I and II. The method that gave the best prediction error was the cross validation, followed by the method of Garcia and Almeida. The prediction error for the extension of the method of Judd and Mees is comparable to that of Garcia and Almeida.

We compared the computation time required for finding the reconstructions in Table III. The extension of the method of Judd and Mees is the fastest and the cross validation is the second slowest. The method of Garcia and Almeida is the slowest even if we use the efficient box-assisted method for finding neighbors given in Ref. [23]. Calculating the nearest neighbors seems a bottleneck.

We also compared these methods with data from the Lorenz96 I model [24,25]. This model is a toy model of the atmosphere, which is defined as

TABLE III. Computation time required for finding the reconstructions for coupled Rössler systems.

Method	Computation time (s)
Boccaletti *et al.*	143.86
Extension of Judd and Mees	0.84
Cross validation	1385.1
Garcia and Almeida	37486.0

TABLE IV. Chosen delays for Lorenz96 I model.

Method	Chosen delays
Boccaletti *et al.*	$z_1(t-1)$, $z_2(t-1)$, $z_3(t-1)$, $z_4(t-1)$, $z_5(t-1)$, $z_6(t-1)$, $z_7(t-1)$, $z_8(t-1)$, $z_9(t-1)$, $z_{10}(t-1)$
Extension of Judd and Mees	$z_1(t-1)$, $z_1(t-3)$, $z_1(t-5)$, $z_1(t-7)$,$z_1(t-9)$, $z_2(t-1)$, $z_2(t-4)$, $z_7(t-4)$,$z_8(t-9)$, $z_9(t-1)$, $z_9(t-10)$, $z_{10}(t-2)$, $z_{10}(t-3)$, $z_{10}(t-5)$, $z_{10}(t-6)$, $z_{10}(t-7)$, $z_{10}(t-10)$
Cross validation	$z_1(t-1)$, $z_1(t-2)$, $z_1(t-3)$, $z_1(t-5)$

$$\frac{dz_i}{dt} = z_{i-1}(z_{i+1} - z_{i-2}) - z_i + F \tag{11}$$

for $i=1,\ldots,n$, where we used a periodic boundary condition $z_{i-n}=z_{i+n}=z_i$ and the parameters $n=10$ and $F=8$. We observed all the variables every 0.05 unit time and generated a time series $\{(z_1(t),z_2(t),\ldots,z_{10}(t))\}_{t=1}^{10\,000}$ of length 10 000. We used the first 9500 points for modeling and the remaining points for evaluating the prediction. We made a model that predicts $z_1(t)$ using the previous observations up to time $(t-1)$. For the proposed methods, we used the maximum delay of 10. For the analysis for the Lorenz96 I model, we do not use the method of Garcia and Almeida because it is too time consuming.

The results are summarized in Tables IV and V. In this example, the two proposed methods have smaller prediction errors than that of Boccaletti *et al.* [16].

We also tested the robustness of the methods against noise. We added 5% Gaussian observation noise to the data of the Lorenz96 I model. We found the reconstructions and the predictive models from the noisy data and evaluated the prediction errors using the clean data.

We show the results in Tables VI and VII. The method of Boccaletti *et al.* is robust in the sense that the performance did not get worse even in the noisy case. The prediction error of the cross validation doubled. But still the two proposed methods enjoy smaller prediction errors than that of Boccaletti *et al.*

This example shows that even under noisy cases, the proposed methods work better.

The proposed methods can give not only better prediction but also better description. Since they provide delays that are specific to an observable, we can find causal relations among observables. The other methods mentioned in this paper cannot do this. The methods for finding a uniform embedding such as those of Cao *et al.* [15] and Boccaletti *et al.* [16] cannot find an optimal delay. The methods that try to reconstruct single states for the entire system such as that of Boccaletti *et al.* [16] and that of Garcia and Almeida [17] cannot describe the relations among observables.

TABLE V. Root mean square errors for Lorenz96 I model.

Method	RMSE
Boccaletti *et al.*	0.5285±0.0280
Extension of Judd and Mees	0.2221±0.0040
Cross validation	0.1064±0.0013

TABLE VI. Chosen delays from noisy data generated from Lorenz96 I model.

Method	Chosen delays
Boccaletti *et al.*	$z_1(t-1)$, $z_2(t-1)$, $z_3(t-1)$, $z_4(t-1)$, $z_5(t-1)$, $z_6(t-1)$, $z_7(t-1)$, $z_8(t-1)$, $z_9(t-1)$, $z_{10}(t-1)$
Extension of Judd and Mees	$z_1(t-1)$, $z_1(t-3)$, $z_1(t-5)$, $z_1(t-8)$,$z_2(t-3)$, $z_2(t-6)$, $z_3(t-1)$, $z_3(t-3)$, $z_3(t-6)$, $z_4(t-1)$, $z_4(t-3)$, $z_4(t-7)$, $z_4(t-10)$, $z_8(t-1)$, $z_8(t-9)$, $z_9(t-1)$, $z_9(t-3)$, $z_9(t-10)$, $z_{10}(t-1)$, $z_{10}(t-4)$
Cross validation	$z_1(t-1)$, $z_1(t-2)$, $z_2(t-4)$, $z_9(t-1)$, $z_{10}(t-1)$,

We applied the extension of the method of Judd and Mees to wind data to extract some causal relationships behind them. The wind data we use here were recorded every 10 min at 160 points in Hokkaido, the northern island in Japan, for a month in July 2003. The wind data at each point were originally recorded using the velocity and direction. However, we converted the format into that of the east and north wind by taking projections. Thus each point has two variates.

We picked the point shown with a circle in Fig. 1 and tried to find the delays up to time $(t-1)$ that have some effect on the east wind at that point at time t. Here the maximum delay is 36. Since there were 160 points and each point has two variates, a total of 11 520 candidate delays were prepared. We ordered the points so that points closer to the one with the circle appear earlier in the list. The result is shown in Fig. 1. Many delays of points that are located to the southwest were selected. The westerlies might be related to these delays. Judging from the magnitude of the delays, the delay selected at (143.8,43.4) is not a direct effect. This point and

TABLE VII. Prediction errors for noisy data generated from Lorenz96 I model.

Method	RMSE
Boccaletti *et al.*	0.5254±0.0333
Extension of Judd and Mees	0.2655±0.0040
Cross validation	0.2269±0.0109

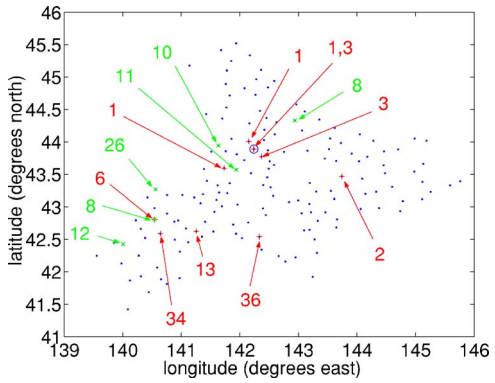

FIG. 1. (Color online) Delays for predicting the east wind at the point with a circle obtained using the extension of the method of Judd and Mees. The dots show points of observation. Symbol +(×) shows points whose east (north) wind helps to predict the east wind at the point with a circle. The numbers shown are the magnitudes of the delays. For example, the number 1 means 10 min.

the predicted point may have the same driving force, such as the winds from the sea and land that occur daily. We also used the cross validation for the same data set with the same maximum delay. The result is shown in Fig. 2. The selected delays are different from those of Fig. 1. This means that similar information can be obtained from other observables. The results for the univariate cases are shown in Table VIII. In the univariate cases, we only used the delays corresponding to the east wind at the point with the circle in Figs. 1 and 2. These residuals are compared in Table IX. The multivariate cases yielded smaller residuals and thus could better explain the causal relationships of the data set. This is not the result of overfitting since the normalized maximum likelihood avoids it by penalizing the number of parameters.

We learned that recently Pecora and Moniz [26] proposed a method for finding a delay embedding from multivariate data. Their method and our methods are not mutually exclusive and we can enjoy benefits from both. If we use first the method of Pecora and Moniz and second one of our proposed methods, then we can initially have a delay embedding and reduce the number of candidate delays using one of our methods. In this case, our methods will be demanded since things are complicated in a network and it is worth picking up only delays that are related to each coordinate. The second approach is to first apply one of our methods and reduce the number of candidate delays and then use the method of Pecora and Moniz. This permutation works well especially if their method takes a long time when the number of candidate delays is huge.

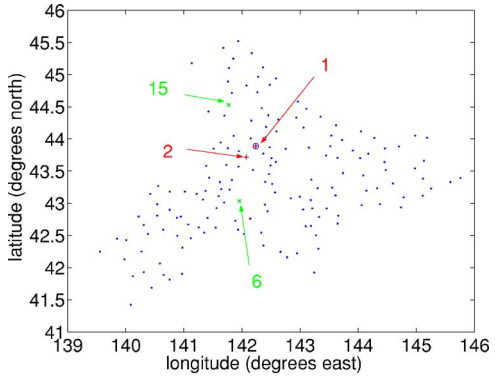

FIG. 2. (Color online) Delays for predicting east wind at the point with a circle obtained using cross validation. The dots show points of observation. Symbol +(×) shows points whose east (north) wind helps to predict the east wind at the point with a circle. The numbers shown are the magnitudes of the delays. For example, the number 1 means 10 min.

TABLE VIII. Chosen delays from the wind data. These delays were for the east wind at the point with a circle in Figs. 1 and 2.

Method	Chosen delays
Extension of Judd and Mees, univariate	1, 3, 5, 12
Cross-validation, univariate	1, 5, 18, 33

In conclusion, we evaluated two algorithms for choosing a nonuniform embedding given a multivariate time series. The first method is an extension of the method of Judd and Mees [10], where we considered multivariate reduced autoregressive models and selected the best set of delays by minimizing the normalized maximum likelihood [19], a variant of the description length. The second method is cross validation. We minimized the prediction error over all candidates for nonuniform embeddings. Using examples, we demonstrated that the proposed methods can find causal relationships among many variables of multivariate data. We also showed that using nonuniform embeddings these methods do better in prediction than uniform embeddings constructed using the method of Boccaletti *et al.* [16]. The proposed algorithms showed better performance than the method of Garcia and Almeida [17]. If we look for accuracy of prediction, we should choose the cross validation, while we should use the extension of the method of Judd and Mees if we want to find a nonuniform embedding within a short time. We also would like to point out here that it is better to reconstruct states for each variable for the following three reasons: we may be able to divide a system into small subsystems by using states reconstructed for each observable; we can enjoy simpler expressions since some variables are not necessary for describing other variables; we can see causal relations among variables. These points are quite important when dealing with complicated systems such as genetic networks, weather forecasting, and the brain since many variables are involved therein.

TABLE IX. Residuals in root mean square error after fitting the wind data.

Method	Residual
Extension of Judd and Mees, univariate	0.6371±0.0011
Extension of Judd and Mees, multivariate	0.6152±0.0018
Cross validation, univariate	0.6412±0.0018
Cross validation, multivariate	0.6346±0.0018

We would like to acknowledge Japan Meteorological Agency for providing the wind data used for producing Figs. 1 and 2. Y.H. also wants to thank Professor Michael Small for stimulating discussions. This study was partially supported by the Industrial Technology Research Grant Program by the New Energy and Industrial Technology Development Organization (NEDO) of Japan, and a Grant-in-Aid for Scientific Research on Priority Areas—Higher-Order Brain Functions (Grant No. 17022012).

[1] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, Phys. Rev. Lett. **45**, 712 (1980).
[2] F. Takens, in *Dynamical Systems and Turbulence*, edited by D. A. Rand and L. S. Young, Lecture Notes in Mathematics Vol. 898 (Springer, Berlin, 1981), p. 366.
[3] T. Sauer, J. A. Yorke, and M. Casdagli, J. Stat. Phys. **65**, 579 (1991).
[4] J. Stark, D. S. Broomhead, M. E. Davies, and J. Huke, Nonlinear Anal. Theory, Methods Appl. **30**, 5303 (1997).
[5] J. Stark, J. Nonlinear Sci. **9**, 255 (1999).
[6] J. Stark, D. S. Broomhead, M. E. Davies, and J. Huke, J. Nonlinear Sci. **13**, 519 (2003).
[7] C. J. Cellucci, A. M. Albano, and P. E. Rapp, Phys. Rev. E **67**, 066210 (2003).
[8] A. M. Fraser and H. L. Swinney, Phys. Rev. A **33**, 1134 (1986).
[9] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, Phys. Rev. A **45**, 3403 (1992).
[10] K. Judd and A. Mees, Physica D **120**, 273 (1998).
[11] A. P. M. Tsui, A. J. Jones, and A. Guedes de Oliveira, Neural Comput. Appl. **10**, 318 (2002).
[12] M. Small, e-print nlin.CD/0312011.
[13] M. Small and C. K. Tse, Physica D **194**, 283 (2004).
[14] S. P. Garcia and J. S. Almeida, Phys. Rev. E **71**, 037204 (2005).
[15] L. Cao, A. Mees, and K. Judd, Physica D **121**, 75 (1998).
[16] S. Boccaletti, D. L. Valladares, L. M. Pecora, H. P. Geffert, and T. Carroll, Phys. Rev. E **65**, 035204(R) (2002).
[17] S. P. Garcia and J. S. Almeida, Phys. Rev. E **72**, 027205 (2005).
[18] J. Rissanen, *Stochastic Complexity in Statistical Inquiry* (World Scientific, Singapore, 1989).
[19] J. Rissanen, IEEE Trans. Inf. Theory **46**, 2537 (2000).
[20] T. Nakamura, K. Judd, A. I. Mees, and M. Small, Int. J. Bifurcation Chaos Appl. Sci. Eng. (to be published). Also see T. Nakamura, Ph.D. dissertation, School of Mathematics and Statistics, The University of Western Australia, Perth, Australia, 2003 (unpublished).
[21] K. Judd and A. Mees, Physica D **82**, 426 (1995).
[22] J. H. Holland, *Adaptation in Natural and Artificial Systems* (MIT Press, Cambridge, MA, 1992).
[23] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis* (Cambridge University Press, Cambridge, U.K., 1997).
[24] E. N. Lorenz, in *Proceedings of the Seminar on Predictability* (ECMWF, Reading, U.K., 1996), Vol. 1, p. 1.
[25] J. A. Hansen and L. A. Smith, J. Atmos. Sci. **57**, 2859 (2000).
[26] L. Pecora and L. Moniz (unpublished).