Minimum description length neural networks for time series prediction

Michael Small* and C. K. Tse
Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

(Received 3 May 2002; revised manuscript received 6 August 2002; published 6 December 2002)

Artificial neural networks (ANN) are typically composed of a large number of nonlinear functions (neurons) each with several linear and nonlinear parameters that are fitted to data through a computationally intensive training process. Longer training results in a closer fit to the data, but excessive training will lead to overfitting. We propose an alternative scheme that has previously been described for radial basis functions (RBF). We show that fundamental differences between ANN and RBF make application of this scheme to ANN nontrivial. Under this scheme, the training process is replaced by an optimal fitting routine, and overfitting is avoided by controlling the number of neurons in the network. We show that for time series modeling and prediction, this procedure leads to small models (few neurons) that mimic the underlying dynamics of the system well and do not overfit the data. We apply this algorithm to several computational and real systems including chaotic differential equations, the annual sunspot count, and experimental data obtained from a chaotic laser. Our experiments indicate that the structural differences between ANN and RBF make ANN particularly well suited to modeling *chaotic* time series data.

DOI: 10.1103/PhysRevE.66.066701 PACS number(s): 02.70.Rr, 05.45.Tp, 05.45.Pq

I. INTRODUCTION

The minimum description length principle states that the model that provides the most compact description of a time series is best. It is an information theoretic incarnation of Ockham's Razor: ‘‘plurality should not be posited without necessity.’’

Estimates of minimum description length (MDL) [1] have been applied to construct radial basis time series models [2]. In fact, it is easy to see that the technique described in [2] may be applied to any pseudolinear nonlinear model [3]. A generalization of MDL for radial basis models *including* nonlinear model parameters has also been described [4]. Although computationally more expensive, this scheme has been shown to be suitable for modeling a wide range of dynamic nonlinearity from time series data [4–6].

Application of a limited form of MDL for polynomial models was explored by Brown and colleagues [7] and extended to the general situation in [8]. For rapidly sampled systems with low noise it was shown that MDL polynomial models are capable of reconstructing polynomial nonlinearities [8]. However, extrapolation or application to nonpolynomial systems remains poor.

Within the engineering community, a radial basis function network implementation of description length was described recently by Leonardis and Bischof [9]. In contrast to Judd and Mees [2], Leonardis and Bischof start from an overly complex model and selectively prune unneeded functions.

Conversely, neural network analysis is perhaps the most popular tool for modeling nonlinear phenomenon yet application of information theoretic techniques for model selection is not well accepted [10]. Nonetheless, performance of neural networks is notoriously dependent on successful training of the model [11]. Typically, a neural network will consist of a very large number of nonlinear ‘‘neurons’’ (the equivalent of basis functions in the nomenclature of radial basis functions). Often, much to the chagrin of statisticians, the number of neurons, or the number of parameters, will approach or exceed the number of data from which the model is constructed [12]. Parameter estimation for neural networks is therefore extremely nonlinear and occasionally overdetermined [11]. To prevent overfitting one will typically only allow the fitting algorithm to continue for some finite (and relatively short) time, known as the training time. Overfitting is therefore avoided because the model parameter values are not optimal. This inevitably leads to a large number of distinct local minima and one is often unsure that performance for a particular model is typical.

Sporadic applications of information theoretic concepts to address the problem of model selection have appeared in the neural network literature. In 1991, Fogel [13] applied an information criterion introduced by Akaike [14] to estimate the size of neural networks for binary classification problems. However, this approach does not readily extend to time series prediction. We also note that the penalty term of the Akaike information criterion is ‘‘slacker’’ than MDL, therefore the optimal models obtained with this criterion tend to be larger. For time series prediction we have found that this produces excessively large models that still overfit the data. However, we do support the rationale expounded in [13] that the choice of model selection criteria is partly a philosophical one. In practice one often selects the criterion that works best for the given data.

Predictive MDL has been described by Lehtokangas and colleagues [15,16] and implemented for autoregressive [15] and multilayer perceptron [16] networks. Unlike the model selection criterion we introduce here, predictive MDL has a constant cost for each model parameter and is therefore similar to the Bayesian information criteria [17].

Leung and colleagues examined prediction of chaotic time series with radial basis function networks and applied several criteria to determine model size [10]. They concluded that a singular value decomposition based form of cross-

*Electronic address: ensmall@polyu.edu.hk validation performed best for model size selection, and MDL performed extremely badly. Their estimate of MDL appeared to be a decreasing function of model size, with no global minimum. However, this violates the minimum description length principle that there is some optimal finite model size. Therefore, their estimate of MDL was clearly performing poorly [10].

In this paper we suggest an alternative implementation of MDL. We also propose a fitting algorithm that deviates from the standard approach for neural networks. When building a neural network, one typically selects some fixed (large) number of basis functions and initializes the parameters randomly. The weights of the basis functions can then be selected with standard least squares [18]. Nonlinear parameters are then fitted iteratively using a time consuming procedure such as back-propagation [19]. Typically the number of parameters (including both the weights of the individual neurons and nonlinear parameters associated with each neuron) is large and given sufficient time, back-propagation will yield an arbitrarily close fit to the data [20]. The result is a model that is overfit for a particular data set and generalizes poorly: a "brittle" model. To avoid overfitting, back-propagation is typically terminated when cross validation [21] indicates an optimal result. However, the combination of cross validation and back-propagation is time consuming and data intensive. One is usually forced to surrender half the available data for cross validation purposes.

We propose a model fitting algorithm which yields a good solution for any fixed number of model parameters (neurons), and we allow training to proceed until the fit appears to be optimal. We avoid overfitting by constraining the number of neurons in the network to minimize the description length of the model. This leads to neural networks that are often far smaller than those observed in the literature, and dynamic behavior that is both realistic and repeatable. Furthermore, by avoiding both back propagation and cross validation our algorithm is not computationally expensive and utilizes available data efficiently.

Section II describes the minimum description length principle in more detail and derives the expression we use to compute this quantity. Section III discusses artificial neural networks and introduces the modeling algorithm we utilize in this paper. Finally, Sec. IV presents some applications of this algorithm to computational and real time series.

II. DESCRIPTION LENGTH

Consider two parties separated by a communication channel. The first party (Bill) has access to a time series and wishes to transmit the data to the second party (Ben), correct to some *finite accuracy*. One possibility is for Bill to transmit each of the time series values, in succession, to Ben. This will incur a fixed cost related to the required accuracy of the data. Alternatively, if there is structure in the data then Bill may build a model of the data and describe that model to Ben, together with initial conditions and the prediction errors of the model. If the model is a good model for that data then describing the model and the model prediction errors will be more compact than the description of the raw data. Conversely, if the model is poor (produces large errors) or is too large, then the description of the model and the model prediction errors will be large.

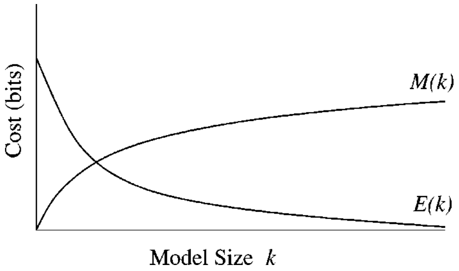

FIG. 1. Description length as a function of model size. The description length of a time series $D(k)$ is the sum of the description length of a model of that time series $M(k)$ and the description length of the model prediction errors $E(k)$. As model size k increases $E(k)$ decreases but $M(k)$ increases. The MDL principle says that the optimal model size is that which minimizes the sum $D(k)=M(k)+E(k)$.

Typically there is a trade off. As model size k increases the model prediction errors decrease—for an optimal model this must be the case. Conversely larger models are more complex and require a lengthier description—this follows from the definition of description length. Let $E(k)$ be the cost of specifying the model prediction errors and $M(k)$ be the cost of describing the model. Intuitively, one can see that $E(k)$ is a decreasing function of k and $M(k)$ is increasing. The description length of $D(k)$ of a given time series utilizing this particular model is then uniquely defined as $D(k)=M(k)+E(k)$ [22]. The minimum description length principle states that the optimal model is the one for which $D(k)$ is minimal. Typical behavior of $E(k)$ and $M(k)$ is depicted in Fig. 1.

Let $\{y_i\}_{i=1}^N$ be a time series of N measurements and let $f(y_{i-1},y_{i-2},\ldots,y_{i-d};\Lambda_k)$ be a scalar function of d variables that is completely described by the k parameters Λ_k. Define the prediction error e_i by

$$e_i=f(y_{i-1},y_{i-2},\ldots,y_{i-d};\Lambda_k)-y_i .$$

Let $\hat{\Lambda}_k$ be the solution of

$$\min_{\Lambda_k}\sum_{i=1}^{N} e_i^2 \tag{1}$$

for a fixed k. For any $\Lambda_k=(\lambda_1,\lambda_2,\ldots,\lambda_k)$ the description length of the model $f(\cdot;\Lambda_k)$ is given by the description length of the k parameters Λ_k [2]:

$$M(k)=\sum_{j=1}^{k}\ln\frac{\gamma}{\delta_j}, \tag{2}$$

where γ is a constant related to the number of bits in the exponent of the floating point representation of λ_j, and δ_j is the optimal precision of λ_j. The precisions δ_j of the optimal

MDL model (for a fixed k) must be computed. Judd and Mees [2] showed that the optimal $(\delta_1, \delta_2, \ldots, \delta_k)$ are given by the solution of

$$\left(Q \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_k \end{bmatrix} \right)_j = \frac{1}{\delta_j}, \tag{3}$$

where

$$Q = D_{\Lambda_k \Lambda_k} E(k), \tag{4}$$

the second derivative of the description length of the model errors $E(k)$ with respect to the model parameters Λ_k.

Rissanen [1] has shown that $E(k)$ is the negative logarithm of the likelihood of the errors $e = \{e_i\}_{i=d+1}^{N}$ under the assumed distribution of those errors:

$$E(k) = -\ln \text{Prob}(e|\Lambda_k).$$

If one assumes that the errors are Gaussian distributed with mean zero and standard deviation σ then

$$E(k) = \frac{N}{2} + \ln\left(\frac{2\pi}{N}\right)^{N/2} + \ln\left(\sum_{i=1}^{N} e_i^2\right)^{N/2}. \tag{5}$$

The assumption of Gaussianity is reasonable in many situations and expedient in all cases. If one has good reason to believe that the distribution of errors should take some other form (such as a uniform distribution if machine precision is the limiting factor) then Eq. (5) may be modified accordingly. For the general case of an unknown distribution of errors the situation is more complex. One alternative is to measure (exactly) the description length of the actual model deviations [using a formulation similar to Eq. (2)]. In the current correspondence we restrict our attention to the situation where the errors are known (or believed) to follow a normal distribution.

In principle we may now compute description length as follows. Solving Eq. (3) yields the precision with which we must specify each parameter. Substituting into Eqs. (2) and (5) one is able to compute the description length of the model $M(k)$ and also of the model prediction errors $E(k)$. We note that the nonlinearity of various model parameters enters into the computation through Eq. (4). For excessively large k a computational bottleneck results from ensuring that the matrix (4) yields a solution to Eq. (3).

III. RADIAL BASIS MODELS ARE NOT NEURAL NETWORKS

Judd and Mees [2] proposed an algorithm to implement the minimum description length principle for radial basis function networks. In this section we introduce the class of neural networks which we will consider in our analysis and contrast these with radial basis networks. We then describe the nonlinear fitting algorithm we employ to solve Eq. (1).

In this section we draw a clear distinction between neural networks and radial basis function networks. Some authors consider multilayer perceptron networks [such as Eq. (7) below] and radial basis function networks to be specific classes of neural networks. In such instances the characteristic common to all ‘‘neural networks’’ is that they are networks (and nothing more) [23]. We do not adopt that nomenclature here, we prefer rather to contrast the two distinct architectures.

A. Radial basis functions and neural networks

Let $z_{i-1} = (y_{i-1}, y_{i-2}, \ldots, y_{i-d})$; a radial basis function network is then a function of the form

$$f(y_{i-1}, y_{i-2}, \ldots, y_{i-d}; \Lambda_k) = \lambda_0 + \sum_{j=1}^{m} \lambda_j y_{i-\ell_j} + \sum_{j=1}^{n} \lambda_{j+m} \phi\left(\frac{\|z_{i-1} - c_j\|}{r_j}\right), \tag{6}$$

where $\Lambda_k = (\lambda_0, \lambda_1, \lambda_2, \ldots, \lambda_k)$, $c_j \in \mathbf{R}^d$, $r_j > 0$ and $1 \leqslant \ell_j < \ell_{j+1} \leqslant d$ are integers. The function ϕ is the radial basis function and is typically Gaussian

$$\phi(x) = \exp(-x^2/2)$$

(a more detailed discussion of other possible forms for ϕ may be found in [4]). The vector c_j is the *center* of the jth basis function and r_j is referred to as the *radius*.

To achieve a fit of Eq. (6) to the time series $\{y_i\}_i$ subject to Eq. (1), one must select the nonlinear parameters c_j and r_j and the linear weights λ_j. The total number of parameters k may be selected subject to MDL.

For functions of the form (6) the procedures described in [2,4] may be employed to find the MDL best model of a time series. In this paper we are interested in the application of description length to neural networks. We restrict our attention to multilayer perceptrons with a single hidden layer [11]. For scalar time series prediction these networks will have d inputs $\{y_{i-1}, y_{i-2}, \ldots, y_{i-d}\}$ fitted to a single output y_i. Mathematically these networks can be expressed as

$$f(y_{i-1}, y_{i-2}, \ldots, y_{i-d}; \Lambda_k) = \xi\left(\lambda_0 + \sum_{j=1}^{m} \lambda_j y_{i-\ell_j} + \sum_{j=1}^{n} \lambda_{j+m} \phi(z_{i-1} \cdot c_j - r_j)\right). \tag{7}$$

For neural networks ϕ is usually selected to be a bounded monotonically increasing function. We choose the hyperbolic tangent

$$\phi(x) = \tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$$

and ξ is another nonlinear function, usually of the same form as ϕ. For time series prediction it has been shown that one only needs to consider the situation where ξ is linear [19]. Furthermore, it is well established that a sufficiently large neural network with a single hidden layer [such as Eq. (7)] is capable of modeling arbitrary nonlinearity [19]. In most cases we find that it is sufficient to set $\xi(x)=x$. However for data that is highly non-Gaussian we have found that choosing ξ such that $\xi(x)$ is Gaussian distributed (mean 0, standard deviation 1) aids the nonlinear fitting procedure.

Unlike many other implementations of neural networks, we have included constant and linear terms explicitly in both Eqs. (6) and (7). This is because we are interested only in the time series prediction problem. Historically and aesthetically one should not resort to nonlinear modeling unless linear methods are inadequate. Therefore, we provide both possibilities and choose that which fits the data best. Typically one expects a combination of linear and nonlinear terms: $m>0$ and $n>0$.

B. Fitting the neural network to the data

The functional forms (6) and (7) are similar and one may suspect that the model selection algorithm should proceed in a manner similar to [2]. Certainly, provided one can compute Eq. (4) and solve Eq. (3), the estimation of description length is no different. However, we are still faced with the problem of fitting the various linear and nonlinear model parameters, and determining (recursively) the optimal model of size k. For this purpose we extend the algorithm previously described for Eq. (6).

(I) Let $\Theta^{(0)}=\{1,[y_{i-1}]_i,[y_{i-2}]_i,\ldots,[y_{i-d}]_i\}$ be the set of all possible constant and linear terms, let $\Phi^{(0)}=\emptyset$ be the empty (null) matrix and let $k=0$. In what follows $\Phi^{(k)}$ is a matrix consisting of the evaluation of the k (selected) neurons and affine terms on the data.

(II) Compute the weights $\Lambda_k=[\lambda_1\,\lambda_2\cdots\lambda_k]$ such that $e=y-\Lambda_k\Phi^{(k)}$ is minimal. Initially, Λ_k is empty and $e=y$.

(III) Generate a set of candidate nonlinear neurons $\Theta^{(k)}$ such that $\Theta^{(k)}\subseteq\{\phi(x\cdot c-r)|c\in\mathbf{R}^d,\ r\in\mathbf{R}\}$ (i.e., choose a set of candidate centers c and radii r).

(IV) Select $\theta\in\Theta^{(0)}\cup\Theta^{(k)}$ such that $|\Sigma_i\theta(y_i)e_i|$ is maximal (i.e., choose the basis function that fits the current error best).

(V) Let

$$\Phi^{(k+1)}=\begin{bmatrix}\Phi^{(k)}\\ [\theta(y_{d+1})\,\theta(y_{d+2})\cdots\theta(y_N)]\end{bmatrix}.$$

(VI) Compute the weights Λ_{k+1} such that $e=y-\Lambda_{k+1}\Phi^{(k+1)}$ is minimal.

(VII) Given $\Phi_{k+1}=[\phi_1\phi_2\cdots\phi_{k+1}]^T$ find i $(1\leqslant i\leqslant k+1)$ such that

$$\left|\sum_{\ell}\phi_i(y_\ell)e_\ell\right|<\left|\sum_{\ell}\phi_j(y_\ell)e_\ell\right|$$

for all j $(1\leqslant j\leqslant k+1)$ (i.e., find the term in the current model that contributes the *least*).

(VIII) If $i=k+1$ then increment k, otherwise set

$$\Phi^{(k)}=[\phi_1\phi_2\cdots\phi_{i-1}\phi_{i+1}\cdots\phi_{k+1}]^T,$$

where ϕ_j is the jth row of the $(k+1)\times d$ matrix $\Phi^{(k+1)}$ (i.e., if the last neuron added *now* contributes the least then enlarge the model, otherwise remove the neuron that does the least).

(IX) If necessary, recompute the weights Λ_k and the model prediction errors e.

(X) Compute the description length $M(k)+E(k)$. If we have reached the minimum then stop, otherwise go to step (III).

The major distinction between this algorithm and that proposed in [2] is that the candidate basis functions (neurons) are recomputed for each model expansion. By expending this additional effort [step (III)] all candidate functions are much better fits to the current model error.

The least mean square estimates of the linear basis function weights are computed in three different places in this algorithm [steps (II), (VI), and (IX)] and for each value of k. Although this calculation is not overly expensive it can be minimized by utilizing a QR factorization [18]. This also aids in the computation of Eq. (4).

Step (IV) selects from amongst the current host of candidates the best fit to the current error, and step (VII) rejects the current worst neuron in the model. Only when the neurons selected in steps (IV) and (VII) differ does the model expand. This helps with the nonlinearity of the problem. Often a combination of two basis functions, *neither* of which are the best fit to the current error, provide a good fit to the error. If the resultant model was found to still be ill-fitting, deeper recursion may be implemented. In all our numerical calculations we found that this single level of recursion was sufficient. We note in passing that an optional step (VIIIa) could be added to apply back propagation (or some similar procedure) to further optimize the parameters of the model of size k. However, the computational cost of such an addition could be substantial.

We have not yet described how the candidate basis functions are generated in step (III). This step is extremely important and the procedure outlined in [2] is not sufficient. When one considers sigmoidal functions for $x\gg1$, $\phi(x)\approx1$ and for $x\ll-1$, $\phi(x)\approx-1$. Therefore, the region of interest is $x\in[-1,1]$, and we choose c and r so that $\{z_i\cdot c-r|i=d+1,d+2,\ldots,N\}\cup[-1,1]\neq\emptyset$. To achieve this we select c such that $\langle z_i\cdot c\rangle_i\in[-1,1]$. The offset term r may then either be selected randomly (for excessively large problems) or computed via a nonlinear optimization routine. For a moderate number of basis functions we have found that a standard Newton-Rapheson steepest descent algorithm [18] rapidly converges to a local minimum and provides significantly improved results.

C. Why the architectures are different

We have already observed that the formulas (6) and (7) are very similar—one simply replaces $(\|z-c\|)/r$ with $z\cdot c-r$. However, there are some fundamental difference between radial basis functions of the form (6) and neural networks (7). As we have described in the previous section, the

TABLE I. Estimates of correlation dimension for data and models. The correlation dimension is shown for the data, iterated model predictions of the same length, and 50 noisy simulations. For the five data sets described in Sec. IV (all contaminated with either experimental or artificial noise) we show the length of time series N, embedding parameters used to estimate the correlation dimension (d_e and τ), MSE of the nonlinear model σ^2, number of linear (m) and nonlinear (n) parameters of the optimal model, and correlation dimension d_c estimates. The correlation dimension estimates are shown for the time series data (''data''), a model simulation (''model''—an iterated model prediction with no noise), and mean and standard deviation of 50 noisy simulations (''surrogates''). The noise level is either 20% or 50% of the model MSE.

									d_c (surrogates)	
System	d_e	τ	N	σ^2	m	n	d_c (data)	d_c (model)	$0.2\sigma^2$	$0.5\sigma^2$
Ikeda	2	1	500	0.035 451	0	25	1.4207	0.039 667	1.3409 ± 0.064 116	1.2388 ± 0.080 872
Rössler	3	1	500	0.3198	2	12	1.6029	1.4763	1.5191 ± 0.050 691	1.6316 ± 0.065 429
Sunspots	2	1	271	1.167	1	14	1.3597	0.918 89	1.1394 ± 0.171 34	1.1047 ± 0.279 47
Infant respiration	4	5	1200	0.057 865	1	12	2.3799	1.7539	2.0441 ± 0.36466	1.9497 ± 0.58539
Chaotic laser	3	1	1000	0.015 707	2	145	1.616	1.6605	1.6207 ± 0.13326	1.6595 ± 0.1312

fitting procedures one typically applies differ tremendously. Generally, the parameters of a radial basis function are fitted using a surface approximation paradigm. Conversely, neural networks are typically ''trained'' using procedures either inspired or motivated by neurobiology. Although we do not use these standard back propagation and training techniques here, we do find that new techniques (described in Sec. III B) are required to fit neural networks well.

The reason for this is related to a more fundamental distinction between the two types of functions. Because neural networks are formed from the dot product $z\cdot c$, a single basis function in $\mathbf{R}^d$ for $d>1$ will always have divergent measure [24]. Conversely, for radial basis functions

$$\left(\int_{\mathbf{R}^d}\left|\phi\left(\frac{\|z-c\|}{r}\right)\right|dz\right)^{1/p}<\infty$$

[if $r>0$ and as usual $\phi(x)\rightarrow 0$ as $x\rightarrow\infty$]. In other words, neural networks are composed of infinitely long ''ridges'' in $\mathbf{R}^d$ and radial basis functions are finite ''bumps.'' Each radial basis function has only local effect, while every neuron (for $d>1$) is global (this is independent of ones choice of ϕ and depends only on the form of the argument). Note that, the convergence theorems obtained for neural networks therefore do not necessarily extend to radial basis function networks, nor vice versa. Specifically, application of minimum description length for radial basis function modeling does not imply that the same method will work for neural networks. As we have found, different parameter estimation procedures are required.

Furthermore, as a consequence of the localization of radial basis functions and non-localization of neurons, every radial basis function network can be simulated by some (multilayer perceptron) neural network, but the converse is not always true [23]. Therefore, the class of functions that can be fitted to arbitrary precision by Eq. (6) is a subset of those that can be fitted by Eq. (7). We must stress that this result only applies when one requires *arbitrary precision*. For most real applications to time series data this is neither required nor necessarily desirable. There are numerous examples of applications for which radial basis models have out-performed neural network methodologies (and of course, vice versa). Specifically, for time series prediction, since $\phi(x)\rightarrow 0$ as $x\rightarrow\infty$ for radial basis functions, one can ensure that trajectories (iterated predictions) of Eq. (6) will remain bounded. The same is not true for neural networks. However, the loss of stability of neural networks appears to be balanced by our computational results in Sec. IV showing superior approximation for chaotic dynamical systems.

IV. APPLICATION TO TIME SERIES DATA

In this section we present the application of this algorithm on two test systems (Sec. IV A) and three experimental data sets (Sec. IV B). The test systems we consider are the Rössler system and the Ikeda map, both with the addition of dynamic noise. We then describe the application of this method to experimental recordings of infant respiration [25], the famous sunspot time series [26], and the chaotic laser data utilized in the 1992 Santa Fe time series competition [11,27].

To provide a quantitative comparison to previous published results we compute mean square error (MSE) and normalized mean square error (NMSE). The MSE is given by $\langle e_i^2\rangle$. Similarly, NMSE is $\langle e_i^2\rangle/\sigma^2$ where σ^2 is the actual variance of the data (over the predicted range). However, as we will demonstrate later, MSE and NMSE are not very good measures of how well the model captures the dynamics. To quantify the structure of the data and that of the model we compare estimates of correlation dimension [28] for the data and model simulations (both with and without stochastic perturbations). By repeatedly generating iterated noise driven predictions, one obtains an ensemble of possible trajectories, and a corresponding ensemble of correlation dimension estimates. As described in [4] these may be considered as a nonlinear surrogate hypothesis test [29] and we quote the mean and standard deviation of the correlation dimension estimates for the surrogates. Table I summarizes these results.

A. Computational experiments

Before considering systems with unknown dynamics and noise we examine the behavior of this modeling algorithm on time series generated from two well known dynamical systems.

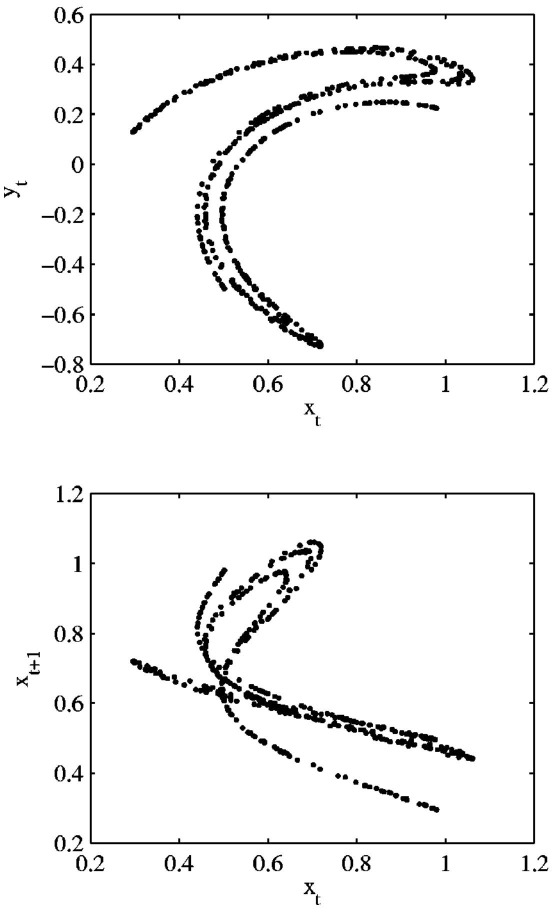

FIG. 2. The Ikeda map. Five hundred points of the Ikeda map are shown, both in original coordinates (x_t, y_t) (upper panel) and reconstructed coordinates (x_t, x_{t+1}) (lower panel). Reconstruction of this map (lower panel) is considerably less trivial than the equivalent reconstruction for either the logistic or Hénon maps.

1. The Ikeda map

The simplest test we apply is reconstruction of the Ikeda map with the addition of various levels of dynamic noise. By *dynamic* noise we mean that system noise is added to the dynamics *prior* to prediction of the succeeding state. The Ikeda equations are given by

$$x_{t+1} = 1 + \mu(x_t \cos\theta_t - y_t \sin\theta_t),$$

$$y_{t+1} = 1 + \mu(x_t \sin\theta_t + y_t \cos\theta_t),$$

$$\theta_t = 0.4 - \frac{6}{1 + x_t^2 + y_t^2}.$$

The bifurcation parameter $\mu = 0.7$ provides chaotic dynamics. We choose to examine this map because the reconstructed dynamics are much more complex than those of either the logistic map or the Hénon map. Figure 2 shows 500 points of this map both in original and reconstructed coordinates. For trajectories of length 500, in the absence of noise, the modeling algorithm was able to accurately reproduce the delay reconstructed attractor.

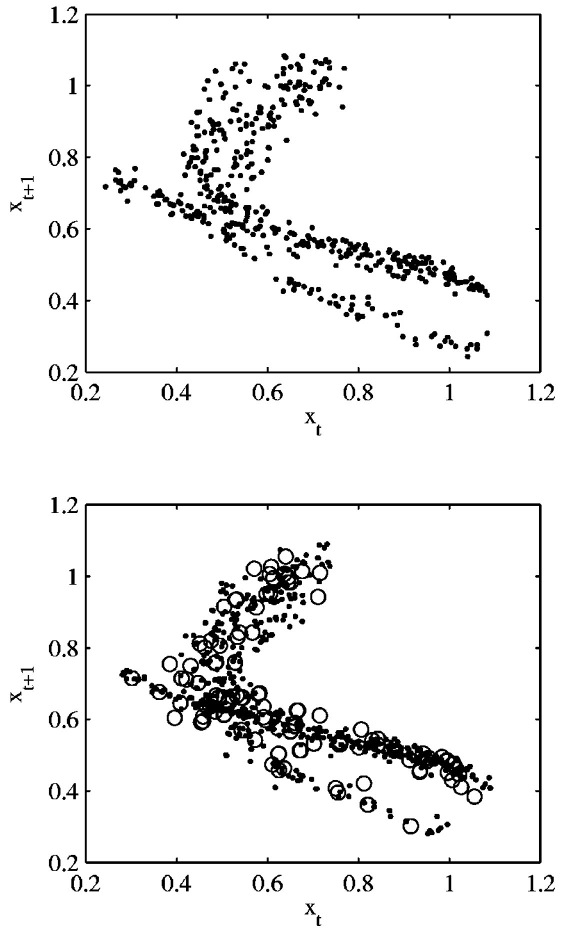

FIG. 3. Reconstruction Ikeda map from a noisy trajectory. The noisy model data (top panel) and the iterated noise free prediction from a model of these data (lower panel, solid dots) are shown. The model generated dynamic behavior exactly equivalent to the original data. Even in the absence of noise the attractor produced from the model prediction is as "noisy" as that from the noisy data. The expected noise free dynamics are shown in Fig. 2. Comparing in-sample and out-sample predictions, this model appears to overfit the data. A second model which does not overfit the data (in-sample and out-sample MSE: 1.25×10^{-3} and 2.15×10^{-3}) produced a high order periodic orbit (lower panel, open circles). The two models contained 91 and 25 nonlinear "neurons," respectively. Correlation dimension estimates for the overfit model were equivalent to the data; for the smaller model these estimates are shown in Table I.

In virtually all real systems deterministic dynamics are corrupted by some dynamic noise (noise that is intrinsic to the dynamics rather than added afterwards). We therefore repeat the Ikeda simulation with the addition of Gaussian random variates to each scalar component at each iteration. The standard deviation of the variates are set at 10% of the the standard deviation of the data. For this level of noise Fig. 3 shows the attractor reconstructed from the original data and attractors reconstructed from noiseless model simulations of length 500. One can see that from this short and noise section of data the basic features of the attractor have been extracted. Table I reveals that the noise free trajectory is a stable limit cycle. However, the limit cycle does lie on the true attractor (Fig. 3). Furthermore, the addition of dynamic

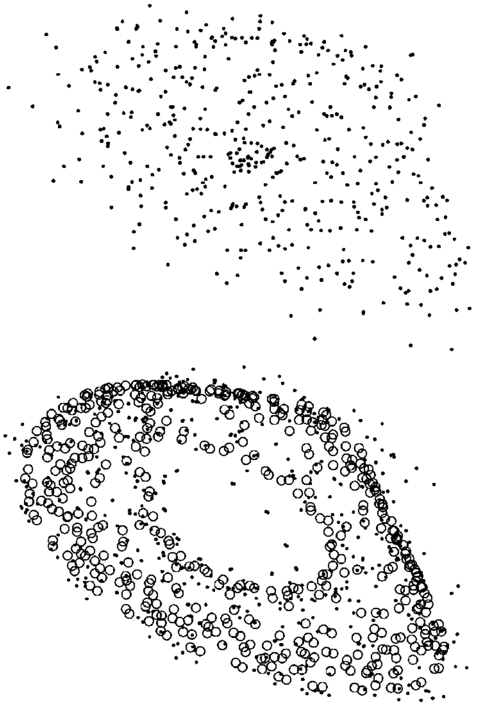

FIG. 4. Reconstruction Rössler system from a noisy trajectory. The noisy model data (top panel) and the iterated noisy free prediction from a model of these data (bottom panel solid dots) are shown. Also shown in the bottom panel is a comparable noise free simulation of this system (open circles). The model generated dynamic behavior equivalent to the differential equations in the absence of noise (Table I). In-sample and out-sample MSE (0.101 and 0.250) were comparable.

noise produces trajectories very similar to the original data. With models built from larger data sets, or lower noise levels the quality of the simulations improves and one obtains the expected chaotic dynamics.

2. The Rössler system

The second computational simulation we wish to consider is a chaotic flow. The Rössler system is given by

$$\dot{x} = -y - z,$$

$$\dot{y} = x + ay,$$

$$\dot{z} = b + z(x - c).$$

For $a = 0.398$, $b = 2$, and $c = 4$ the system exhibits ‘‘single-band’’ chaos. We integrated these equations with step size 0.5 adding dynamic noise to each component at each step (and then using the noisy coordinates for the integration to the next time step) to generate 500 points of the system. From this noisy data we constructed a neural network model (the optimal model had only 12 ‘‘neurons’’) using the methods described in Sec. III. Figure 4 shows the original embedded data, the reconstruction from a noise free simulation of the model and the clean attractor (computed in the absence of noise). One can clearly see that the main features of this chaotic system are reproduced from the model of this short and noisy time series segment. Table I confirms that the correlation dimension estimates for the data and surrogates were comparable. Underestimation of correlation dimension for this data (and possibly the experimental systems in the next section) is due to the finite short and noisy time series. Larger, noise free data produced more accurate estimates. Irrespective of this, the importance of Table I is as a *comparison* of statistic values [5]. With higher noise levels or shorter time series we found that the reconstructed dynamics did not satisfactorily mimic the true behavior. For longer or less noisy data segments we found that performance improved.

B. Experimental data

We now present the application of this algorithm to data from three experimental systems: the Wolf annual sunspot time series [25], experimental recordings of infant respiration [26], and the chaotic laser data utilized in the 1992 Santa Fe time series competition [11,27]. We have deliberately selected these three sources of data because each of them has been the focus of considerable attempts to model the dynamics.

1. Sunspots

The annual sunspot count time series has been the subject of substantial interest in both the physics and statistics communities. Tong [26] describes models of this time series using both autoregressive (AR) and self-exciting threshold autoregressive (SETAR) models. Judd and Mees have subsequently shown that superior predictive performance can be achieved with nonlinear radial basis function models [2,31].

For fairness of comparison we transform the raw data according to

$$y_t \mapsto 2\sqrt{y_t + 1} - 1. \tag{8}$$

This transformation was selected by Tong to improve the performance of linear models with this highly non-Gaussian distributed data. Table II compares the mean sum of square prediction error achieved with our algorithm and the methods proposed by Tong [26] and Judd and Mees [2].

Our main interest is not in the MSE, but in dynamic performance. Linear models described by Tong [26] behave as a stable focus. The nonlinear methods described by Judd and Mees produce either stable foci [2] or a stable periodic orbit [31]. Figure 5 shows that the algorithm described here generates chaotic dynamics that closely resemble the original time series. In Table I we observed fractional correlation dimension in both the data and model simulations. Furthermore, prediction over a longer time horizon shows that the methods described here perform better than the alternatives.

The iterated model prediction shown in Fig. 5 exhibits dynamics remarkably similar to those observed in the historical data. By comparison, linear models clearly cannot capture the long term (nonlinear) dynamics. Radial basis models described in [2,31] exhibit stable periodic orbits and only

TABLE II. Mean free run prediction error for the annual sunspot count time series. Mean sum of the square of the prediction error for models of the sunspot time series produced with five different modeling techniques are reported. Values marked with * are those reported in [2]; † denotes a typical result from an equivalent model employing the methods described by [2] (we do not have access to the earlier model). All other results are computed directly. Following Tong [26], the AR(9) and reduced AR (RAR) models are computed from the untransformed data. Applying the transformation (8) produced similar results. The SETAR model is described in [30].

	MSE	
Model	1980–1988	1980–2002
AR(9)	334*	416
SETAR	413*	1728
RAR	214*	291
Radial basis	306*	489†
Neural network	625	356

approximate the true dynamics when driven by high dimensional noise. Although the predictions of Fig. 5 are qualitatively plausible, we do not claim them to be quantitatively accurate. Table II clearly shows that the neural network model behaves poorly for short term prediction. We are certainly not claiming to predict the observed values for the entire first half of this century.

2. Infant respiration

Radial basis models built with minimum description length have previously been tested with infant respiratory data [4]. These data are recordings of instantaneous abdominal cross section of human infants during normal sleep. It has been shown that these data are not consistent with a linear noise process [25] and that the data are *consistent with* the hypothesis of deterministic chaos. However, nonlinear radial basis models of these data typically behave as a noise driven periodic orbit [4]. While this result is consistent with the conclusions of [25], it is perhaps unsatisfactory that the only deterministic structure that one may extract from these data is a periodic motion. It has recently been observed that in certain circumstances complex period doubling phenomenon may be used to describe this data [32]. This is an attractive observation, but the phenomenon has not been observed consistently in all such data using the methods described in [32].

In this section we apply the modeling algorithm described in this paper to several recordings of infant respiration. In each case we find that the MDL best model of this data exhibits chaotic dynamics and the free run behavior of the model behaves qualitatively similar to the data. Figure 6 depicts a representative data set and simulation. In Table I we see that correlation dimension for *noisy* simulations are comparable to the data (but not without noise). Perhaps this is further evidence of the stochastic behavior left unmodeled by our algorithms (a similar observation was made in [25]).

3. Chaotic laser dynamics

Our final test system is data from a chaotic laser experiment. This data was utilized in the 1992 Santa Fe time series competition. From a large number of potential modeling regimes a nearest neighbor technique [27] and a neural network model were found to perform best [12].

We utilize the same data as described in [12,27] to build a model using the algorithm described in Sec. III. Initial model results were relatively poor. We found that transforming the data so that it was normally distributed prior to modeling

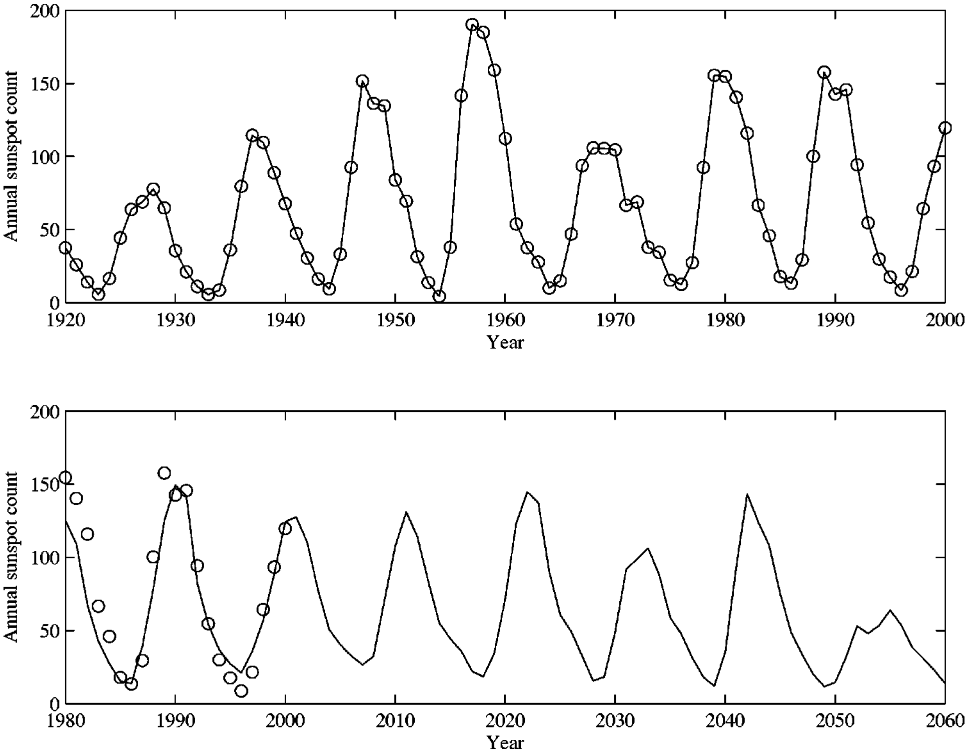

FIG. 5. The annual sunspot count. The actual sunspot count and iterated predictions from a neural network model of these data (the model is described in Table I). The top panel shows the actual sunspot count for each year of the period 1920–2000. The bottom panel shows a noiseless free-run (iterated) prediction for the model of these data over a period of 80 years (1980–2060). Actual known values are also shown (circles) for the years 1980–2000. The free run prediction does not fit the dynamics exactly, but it does provide a good model of the dynamics. Qualitative features are common to both panels. The annual sunspot count is a dimensionless quantity derived from the number of sunspots observed throughout that year (see [26]).

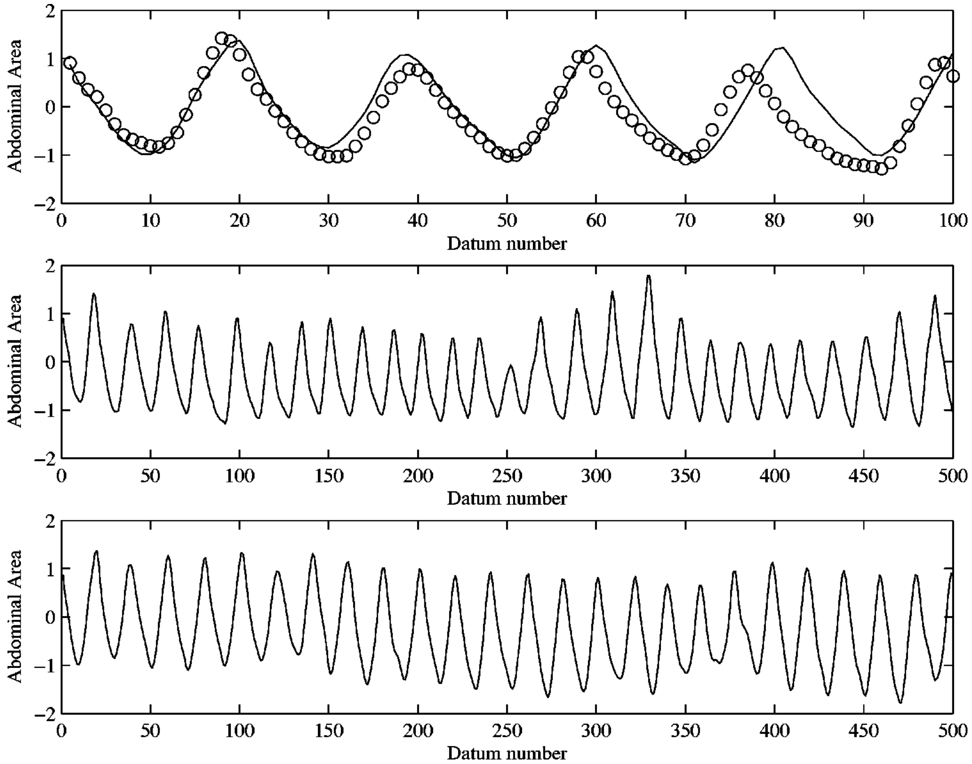

FIG. 6. Human infant respiration. The top panel shows the short term prediction from infant respiration data. True data are depicted as circles; the model prediction is a solid line. The second and third panels show longer representative segments of both the original data (second panel) and the MDL-best neural network model (Table I) free-run prediction. The model is built from the data shown in the second panel and the free-run prediction is generated without any additional noise. In this example the model simulation does not exhibit the peak variation present in the data. All other features are comparable: the same irregular asymmetric wave form and frequent variation in amplitude are present in both the data and simulation, and both had correlation dimension exceeding 2. The measured abdominal area is proportional to the cross sectional area, but the units of measurement are arbitrary and have been rescaled to have a mean of zero and a standard deviation of one [25].

produced far superior results. We believe that this data is sufficiently non-Gaussian so that the assumption that $\xi(x)=x$ in Eq. (7) is inappropriate. We are forced to impose an arbitrary transformation to aid the modeling algorithm and improve results. In Table III we quantitatively compare our prediction results to those presented in [12,27].

Figure 7 depicts representative results of the modeling algorithm with the inclusion of this static nonlinear transformation of the data. We note that the qualitative behavior of this model is comparable to the best modeling results of Sauer [27] and Wan [12]. Sauer's model utilized a nearest neighbor technique [27] and therefore does not provide an actual estimate of the equations of motion (one cannot differentiate a nearest neighbor prediction). In [12] user intervention is required to produce a plausible prediction. The dynamic behavior depicted in Fig. 7 is produced directly from the data. The NMSE of our modeling algorithm is not substantially better than those of [12,27], but the qualitative behavior is better and the algorithm therefore provides equations of motion that are a more plausible model for the underlying dynamics. Table I confirms that the behavior of data and model simulations (with and without noise) are remarkably similar. Perhaps indicating that this system (unlike that in the previous section) is completely (or largely) deterministic.

The model prediction in Fig. 7 is typical of our results. But, we observed that small changes in the initial conditions of a model (less than 0.01% of the data values) greatly changed the NMSE prediction error over the next 100 data. Using the expected values we were able to optimize the initial condition of the model and obtained improved "predictions." Of course, these are not true predictions as they require knowledge of the actual trajectory. Rather, we are providing a maximum likelihood estimate (MLE) of the ini-

TABLE III. Mean free run prediction error for the chaotic laser data. NMSE for models of the laser time series produced with three different modeling techniques are reported. NMSE are computed for 100 point free run predictions initiated at datum number 1000, 2180, 3870, 4000, and 5180 (these initial conditions are those selected in [27]). Values marked with * are those reported in [27].

	NMSE				
Model	1000	2180	3870	4000	5180
Sauer [27]	0.027*	0.065*	0.487*	0.023*	0.160*
Wan [12]	0.077*	0.174*	0.183*	0.006*	0.111*
MDL-neural network	0.066	0.061	0.086	0.479	0.038

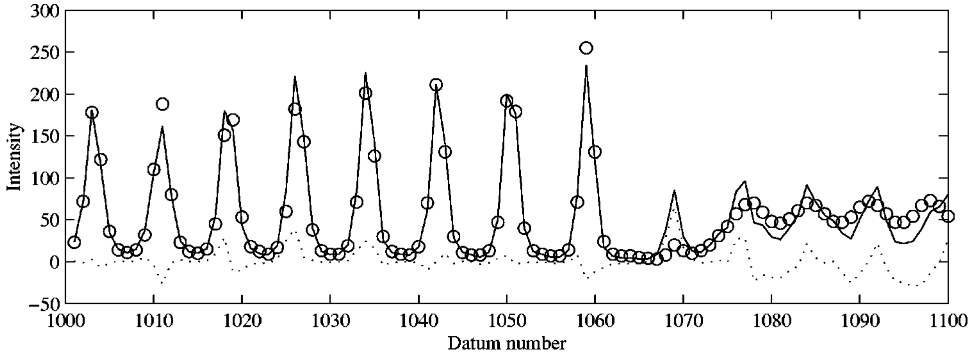

FIG. 7. Chaotic laser dynamics. Free-run prediction (solid line) and actual data (circles) for a MDL-best model of the chaotic laser data. The prediction error is shown as a dotted line (NMSE of 0.0955). The simulated performance is qualitatively comparable to those presented in [12]. However, the neural network model described in [12] "breaks down" soon after the collapse of the laser. We found that the neural network model was large and highly chaotic; tiny changes in the initial conditions yielded substantial variation in the model predictions. This model evidently exhibits extremely sensitive dependence on initial conditions and the uncertainty of this simulation as a prediction is therefore great. The laser intensity takes values from 0 to 255 (i.e., the units of measurement are arbitrary).

tial state given the observed trajectory. For that MLE value we compute a model *simulation*, the deviation between the actual observed value and the MLE of the initial state is substantially less than the coarse grain digitization of the model. Therefore, initial states that are indistinguishable to the experimental apparatus exhibit wildly varying performance. Such variation in prediction draws into doubt the significance of the relative small quantitative difference observed in the NMSE depicted in Table III. We prefer only to conclude that the model produced by this algorithm provided quantitatively realistic simulations, which for the correct choice of initial condition could shadow the true trajectory.

V. CONCLUSION

Neural networks have a happy history of producing good (and sometimes not so good) results in situations where the number of parameters exceeds the number of available data ([12] provides a good example of both cases). However, this is not a contradiction of the statistical view that N data points may only be used to fit (at most) N parameters. The important consideration is the precision with which one chooses to specify each parameter. Assuming *infinite* precision of every observation and parameter, a (linear, linearly independent) problem with N observations is overdetermined if the number of parameters k is less than N. Conversely if $k \geqslant N$ the problem is underdetermined and one can achieve an arbitrary fit to the data. By terminating the training of a network before optimization one obtains parameters with a relatively low precision and one is therefore able to specify a large number of them $k > N$. However, because parameter optimization is a nonlinear problem this premature termination leads to a local minima—very often repeated application will yield a different local minima and different model behavior. One then simply chooses the model that performs best on the training data.

In an attempt to address this problem, predictive MDL [15,16] and other information theoretic model selection criteria [13] have been suggested in the literature. However, none of these techniques consider the *precision* with which parameters must be specified. Because the MDL criterion described here computes the precision of the parameters one has a much fairer estimate of the best model of a particular data set. Furthermore, this avoids the need to waste (often rare or valuable) data during cross validation [10].

We cannot prove that this algorithm will work best for any given data set. For any particular data set we actually expect this algorithm to be sub-optimal. However, theory shows that the functional form (7) is adequate for any nonlinearity and, with sufficiently large d and k, it will capture the dynamics of a sufficiently long time series. Assuming the time series *is* sufficiently long (let N_0 be the minimum such length, so that $N > N_0$ is sufficient), then there exists d_0 and k_0 such that the neural network (7) captures the required nonlinearity. We rely on heuristic techniques to determine d_0 and MDL selection criteria is used to find k_0. If $N < N_0$ then we have insufficient data to find the optimal model with this approach. In some situations other modeling algorithms may perform better: for example, a global polynomial model may model polynomial nonlinearities well. However, for $N < N_0$ MDL selection will still find the best model size $k(d,N)$ *given the available data.*

To justify our failure to find the optimal model in every case, we note that the combinatorial nature of this problem means that there is no known polynomial time algorithm to find a solution. One need only note that a restricted version of our MDL nonlinear model selection problem can be recast as the knapsack problem [33]. We therefore conclude that it is highly unlikely that an efficient *generic* algorithm exists for estimating the best neural network (or basis function) model of a given data set.

It is interesting to note that many of the modeling results presented here (most notably those of the chaotic laser dynamics) exhibit the (expected) sensitive dependence on initial conditions. This sensitivity is sufficient to generate a wide variety of dynamic evolution within the experimental precision of the raw data. The data are digitized as 10-bit integers. However, change in initial condition of less than 0.001 (in each component) provided indistinguishable initial conditions but NMSE over the prediction range of 100 values varies between the optimal results shown in Table III and NMSE greater than 1. Therefore, if this model is an accurate representation of the dynamics in question, then comparing NMSE over this horizon is irrelevant because of the excessive uncertainty in the initial conditions. One should test how well the model captures the dynamics. NMSE (either one step or iterates as in Tables II and III) is a poor measure of dynamic fit. The correlation dimension or other dynamic invariants are far better (see Table I) [5].

We do not emphasize the predictive power of this algorithm. Each of the systems tested was potentially chaotic. We demonstrated for the laser data that prediction from the model was poor because of the sensitive dependence on initial conditions and possible undersampling of the original experiment. However, in each of the experimental systems we found that the qualitative behavior of model simulations was highly accurate. Realistic chaotic dynamics were observed for the sunspot time series. Simulations of infant respirations appear indistinguishable from real data. Finally, simulations from models of the Santa Fe laser data exhibited the same features as the data and achieved (for optimal selection of initial conditions) free-run prediction that exceeded previous results. Comparing dynamic invariants of the data and model simulations showed good agreement (Table I) and provided a fairer and more useful test of ''goodness'' of the models.

Finally, we note that model prediction errors of test and fit data observed for the simulated systems were not exactly equal. We have observed, for MDL-best models a slight, but systematic overfitting of the data. The one-step in-sample MSE values quoted in Figs. 3 and 4 were systematically lower than the corresponding out-sample MSE. This is due to a flaw in the current algorithm. To alleviate computational burden we assumed that the only significant parameters Λ_k were the linear ones $(\lambda_0,\lambda_1, \dots ,\lambda_k)$. While this is clearly only an approximation it seems to produce adequate results. For the case of radial basis models it was found that the additional expense of computing the full description length provided only a slight advantage for the final model [4]. It is likely that the improvement in neural network models afforded by the full calculation would also be marginal.

ACKNOWLEDGMENTS

This work was supported by a Hong Kong Polytechnic University research grant (No. G-YW55). The authors acknowledge the helpful comments of Z. Yang and J. P. Barnard regarding neural network technology.

[1] J. Rissanen, *Stochastic Complexity in Statistical Inquiry* (World Scientific, Singapore, 1989).

[2] K. Judd and A. Mees, Physica D **82**, 426 (1995).

[3] A pseudolinear nonlinear model is a model that is a nonlinear function of the model variables, but only a linear function of the model parameters [18].

[4] M. Small and K. Judd, Physica D **117**, 283 (1998).

[5] M. Small and K. Judd, Physica D **120**, 386 (1998).

[6] M. Small, D. Yu, R. Clayton, T. Eftestøl, and R. G. Harrison, Int. J. Bifurcation Chaos Appl. Sci. Eng. **11**, 2531 (2001).

[7] R. Brown, N. F. Rulkov, and E. R. Tracy, Phys. Lett. A **194**, 71 (1994).

[8] M. Small, K. Judd, and A. Mees, Phys. Rev. E **65**, 046704 (2002).

[9] A. Leonardis and H. Bischof, Neural Networks **11**, 963 (1998).

[10] H. Leung, T. Lo, and S. Wang, IEEE Trans. Neural Netw. **12**, 1163 (2001).

[11] N. Gershenfeld, *The Nature of Mathematical Modeling* (Cambridge University Press, Cambridge, England, 1999).

[12] E. A. Wan, in *Time Series Prediction: Forecasting the Future and Understanding the Past,* Vol. XV of *Studies in the Sciences of Complexity*, edited by A. Weigend and N. Gershenfeld, Santa Fe Institute, (Addison-Wesley, Reading, MA, 1993), pp. 195–217.

[13] D. B. Fogel, IEEE Trans. Neural Netw. **2**, 490 (1991).

[14] H. Akaike, IEEE Trans. Autom. Control **19**, 716 (1974).

[15] M. Lehtokangas, J. Saarinen, and K. Kaski, Appl. Math. Comput. **75**, 151 (1996).

[16] M. Lehtokangas, J. Saarinen, P. Huuhtanen, and K. Kaski, Neural Comput. **8**, 583 (1996).

[17] G. Schwarz, Ann. Stat. **6**, 461 (1978).

[18] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C* (Cambridge University Press, Cambridge, England, 1988).

[19] D. Rumelhart, R. Durbin, R. Golden, and Y. Chauvin, in *Mathematical Perspectives on Neural Networks,* edited by P. Smolensky, P. Mozer, M. Rumelhart, and D. Rumelhart (Lawrence Erlbaum Associates, Hillsdale, NJ, 1996), pp. 533–566.

[20] Because the model is a linear combination of linearly independent functions, this is obvious.

[21] Cross validation is a model selection criteria. The available data is divided into two sections: fit data and test data. The model is then fitted to the fit data and the model selection determined by the performance on the test data. Of course, the test data must be known prior to modeling, and is part of the model building process (one cannot use it for ''honest'' test predictions).

[22] Description length $D(k)$ will depend on the particular time series under consideration and the model selected. Computation of $M(k)$ and $E(k)$ will depend on the particular encoding once chooses for the model and for rational numbers. We use the optimal encoding of floating point numbers described by Rissanen [1].

[23] D. P. Mandic and J. A. Chambers, *Recurrent Neural Networks for Prediction* (Wiley, London, 2001).

[24] S. Ellacott and D. Bose, *Neural Networks: Deterministic Methods of Analysis* (Thomson Computer Press, London, 1996).

[25] M. Small, K. Judd, M. Lowe, and S. Stick, J. Appl. Physiol. **86**, 359 (1999).

[26] H. Tong, *Non-linear Time Series: A Dynamical Systems Approach* (Oxford University Press, New York, 1990).

[27] T. Sauer, in *Time Series Prediction: Forecasting the Future*

and Understanding the Past, (Ref. [12]), pp. 175–193.

[28] D. Yu, M. Small, R.G. Harrison, and C. Diks, Phys. Rev. E **61**, 3750 (2000).

[29] The hypothesis is that the model adequately described the dynamics.

[30] D. Ghaddar and H. Tong, Appl. Stat. **30**, 238 (1981).

[31] K. Judd and A. Mees, Physica D **120**, 273 (1998).

[32] M. Small, D. Yu, and R. G. Harrison, in *Space Time Chaos: Characterization, Control and Synchronization*, edited by S. Boccaletti, J. Burguete, W. G. -V. nas, H. Mancini, and D. Valladares (World Scientific, Singapore, 2001), pp. 3–18.

[33] The knapsack problem is a known NP-complete problem. If a given problem can be recast as an NP complete problem then it too is NP-complete. All NP-complete problems are equivalent and are as computationally difficult as (for example) the traveling salesman problem. Therefore, the problem we consider here has no (known) polynomial time solution.