

On Selecting Models for Nonlinear Time Series

Kevin Judd

Alistair Mees

January 9, 1995

Abstract

Constructing models from time series with nontrivial dynamics involves the problem of how to choose the best model from within a class of models, or to choose between competing classes. This paper discusses a method of building nonlinear models of possibly chaotic systems from data, while maintaining good robustness against noise. The models that are built are close to the simplest possible according to a description length criterion. The method will deliver a linear model if that has shorter description length than a nonlinear model. We show how our models can be used for prediction, smoothing and interpolation in the usual way. We also show how to apply the results to identification of chaos by detecting the presence of homoclinic orbits directly from time series.

1 The Model Selection Problem

As our understanding of chaotic and other nonlinear phenomena has grown, it has become apparent that linear models are inadequate to model most dynamical processes. Nevertheless, linear models remain attractive because of the great power derived from the elegance and completeness of the theory. The art of constructing nonlinear models is by comparison in its infancy, and is unlikely in the near future to develop anything like the completeness that linear modeling possesses. However, there are steps in the process of building a linear model that we would be wise to emulate when building nonlinear models. This paper tries to use some of what is known about linear models to improve the process of building nonlinear models.

The processes of building a linear model has two levels. On the lower level, one fits a given parametrised model to a time series, but there is a higher level where the model is chosen in the first place. The point is that there is always more than one possible model—indeed there are infinitely many—so one must somehow decide which to use before the fitting even starts. One commonly-used criterion for choosing the best model is that it should capture the essential dynamics of the time series without “over-fitting”, which results in including in the model aspects of the time series that should be attributed to noise. We call this higher level of the model building process, the *model selection problem*. There has been a great deal of recent work on algorithms to construct nonlinear models, but much of this work has ignored the important selection problem.

We assume in this paper that all models to be considered are of the following form. There is a random process generating $(Y_t, X_t) \in \mathbf{R} \times \mathbf{R}^d$, $t \in \mathbf{Z}$, for which there is a relationship

$$Y_t = F(X_t) + \epsilon_t, \tag{1}$$

where the random variables ϵ_t are independent of Y_r and X_s for $r \geq t$ and $s > t$. Generally, a strong form of this model is assumed where the ϵ_t are i.i.d. with finite variance. *Building a*

model means, given a realization of this process (y_t, x_t) , $t = 1, \dots, n$, find an approximation \hat{F} of F .

If one is given only a scalar time-series Y_t , then, by the Takens embedding theorem and its extensions [24, 21] one can define

$$X_t = (Y_{t-\tau}, Y_{t-2\tau}, \dots, Y_{t-d\tau})$$

for some lag $\tau > 0$ and embedding dimension d . Other embedding strategies are also possible. If Y_t is a multivariate time-series, then there is a map for each component of Y_t , and the maps can be considered independently. In this paper we take the embedding as given, and we assume that difficult questions such as choice of embedding dimension d and lag τ have been answered already. Whatever embedding strategy is used, the time-series model is linear only when $\hat{F}(X)$ is linear.

Typically, when modeling a time series one restricts attention to a subclass of models $\hat{F}(X) = G(X, \lambda)$, parameterized by $\lambda \in \mathbf{R}^m$ for some m . The potential number m of parameters may be infinite for a class of models, for example, the polynomials or Fourier series. In the context of a class $G(X, \lambda)$, the fitting of a model \hat{F} corresponds to estimation of the parameters λ subject to the constraint that only finitely many components of λ are nonzero. The *model selection problem* asks, given the possibly infinite number of models resulting from the different combinations of nonzero parameters, which is the best?

For linear models, for example auto-regressive models, the parameters are potentially infinite in number but the selection problem is relatively easy. One fits models $\text{AR}(k)$ of increasing order, $k = 0, 1, 2, 3, \dots$, until it is found that increasing the order of the model makes no “significant” improvement. This method is simple, though we show in Section 5.2 that it can be improved.

In the case of nonlinear models, however, there are uncountably many possible subclasses to choose from. Some alternatives to linear models that appear in the literature are global polynomials, piecewise linear functions, orthogonal functions, radial basis models, and neural networks, to name but a few. Unless one has some prior knowledge about the time series, derived, say, from the physics of the phenomenon being studied, it is not at all clear how to select a subclass to work with. Two key developments in recent years seem to provide at least a partial answer. They are

1. the work of Barron and others on the comparative power of various methods for high dimensional functional approximation, and
2. the work of Rissanen on minimum description length models.

In this paper we will discuss a class of models we call *strong pseudo-linear models*. The adjective “strong” refers to development 1 above, while “pseudo-linear” refers to the appearance of parameters in a way that allows easy fitting. A subset selection process will be required in the fitting process and this is described later. We use development 2, minimum description length, to select the most appropriate model from a class of models. We discuss several applications of the optimal nonlinear models obtained by our methods, which include time series prediction and the identification of bifurcation mechanisms from a time series. We will also note that some of our methods can be applied to purely linear models to improve upon them.

1.1 Weak and strong approximations

We discuss the functional approximation problem only briefly. Barron and others [3] have shown that certain methods of function approximation are considerably more powerful than others in high dimensions, in the sense that the optimal approximation error grows far more slowly with dimension for one class than for another. In this paper we shall call methods which behave well with increasing dimension *strong* approximation methods, and others *weak* approximation methods. The strong methods include neural nets, radial basis functions, wavelets and piecewise linear approximations such as tessellation and triangulation [13]. Weak methods include linear approximations, global polynomials, and Fourier transforms. Because we usually require higher dimensionality in our models, we are mainly interested in strong models, though it is appropriate to check whether a weak model would suffice in any given case. We deal with this later by including the possibility of a purely linear model in our automatic model selection procedure.

A characteristic feature of strong models is that the fitting process is inherently nonlinear. Global linear, global polynomial and Fourier models are all linear operations on instantaneous transforms of the data: an invertible linear transformation is usually followed by projection into a subspace chosen by least squares. The resulting ease of use accounts for the popularity of weak models but now that they are provably inferior for general high dimensional problems, we are faced with the problem of how to work with strong models.

The solution used by the neural net community is usually to optimize the model fit by steepest descent (back-propagation) or some related method. The solution given in the present paper is rather different: we use a subset selection method to, in effect, make a linear method behave like a nonlinear one. The details will be discussed later, in Section 3.

1.2 Minimum description length

The model selection problem involves selecting k nonzero elements of the m -vector λ in a given subclass of nonlinear model, $G(X, \lambda)$, for example, a neural net or radial basis function model with a given structure. There are existing methods for tackling the selection problem, but most of them do not allow comparison across model types: say, comparing the neural net with the radial basis function model. That is, can we say whether the optimally selected neural net model is more or less effective than the optimally selected radial basis function model at capturing the essentials of the time series? As stated, this is a badly posed problem, but the intention is that “essentials” means we should prefer methods that are resistant to over-fitting.

One approach that can be used for selection and also, in principle, for model comparison, derives from trying to find a correct statement of our ill-posed problem. It is Rissanen’s use of *minimum description length* [20] to characterize model quality. Rissanen’s argument is that we should regard good models as those that compress the data best: this is, of course, a form of Ockham’s Razor [10]. It turns out to be a very powerful idea, with applications in many areas of data analysis including our present problem.

To measure data compression, we envisage encoding the data for optimal transmission: one way is to use a *two part code*. First we transmit the model, including the values of all of its parameters. The receiver reconstructs the model, and we then transmit enough additional information to allow the receiver to use the model to reconstruct all of the observed data to its full measured accuracy. We say that the total number of bits sent in this way is the

description length of the data under the model, and we look for a model that has minimum description length (MDL) among all models being considered. It is not possible here to give complete details but we shall try to give an outline sufficient to motivate the calculation that leads to a useful approximation to the description length. The reader should note that we are over-simplifying in places, for example in suppressing details of the difference between discrete and continuous distributions. The details are in Rissanen's book [20], from which this outline is adapted.

Suppose, realistically, that the data are all given to finite accuracy, so we can treat them as confined to a countable number of cells in $\mathbf{R} \times \mathbf{R}^d$. If we have a probability distribution P on the data, we can use it to define an optimal encoding of any realization of a time series $Z = \{(Y_t, X_t)\}_{t=1}^n$. A standard coding theory result is that the minimal code length under P is $-\log_2 P(Z)$ bits¹. Our assumed dynamical process (1) defines the distribution of $\epsilon_t = Y_t - F(X_t)$, from which we can find the required distribution $P(Z|F)$ for given F .

For the first part of the two part code, the model transmission, we have to describe \hat{F} where $\hat{F}(X) = G(X, \lambda)$, so we must first transmit a program for calculating G . For simplicity, we assume here that all programs are the same length; this is equivalent to transmitter and receiver agreeing beforehand on a fixed number of model classes, assumed equally likely to be used, so that the transmitter need only send the label of the class actually selected. In this paper we do not consider the full comparison problem so we will not be concerned with most of the details; we do, however, need to be able to compare models with the same functional form G but differing numbers and combinations of nonzero parameters (that is, the number and position of nonzero elements of λ may vary).

Thus to completely specify \hat{F} we must transmit the k nonzero elements of λ , which we can assume (without loss of generality) to be $\lambda_1, \dots, \lambda_k$. This will allow the receiver to construct $P(Z|\lambda)$ and so to decode the Huffman-encoded data that follows in the second part of the two-part code (since, by the assumption about a fixed number of model classes, we are taking $P(Z|\hat{F})$ as identical to $P(Z|\lambda)$). The elements of λ are real numbers, so we will have to truncate them in order to transmit them in a finite code length. The key to the use of the MDL method in our problem is to realize that we can optimize over the truncation.

Rissanen's approach assumes a prior probability distribution on the parameters. (He argues that in fact the final results are not at all sensitive to the choice of prior.) In the absence of other information, we choose the prior corresponding to the computer representation of floating point numbers: roughly speaking, this corresponds to an exponential distribution centered around 0. We choose to send the j th parameter λ_j to a certain relative accuracy δ_j , so we actually send $\bar{\lambda}_j$ which contains only the first $\log_2 \delta_j$ bits in the fractional part of the normalization of λ_j . By considering the floating point representation, we show in Appendix A.1 that the code length needed to specify the parameters $\bar{\lambda}_j$, $j = 1, \dots, k$ is a certain function

$$L(\bar{\lambda}) \approx \sum_{j=1}^k \log \frac{\gamma}{\delta_j}$$

where L is in bits if logarithms are in base 2. The constant γ is not critical and represents the number of factors of 2 required in the exponent of a floating-point representation of a parameter: $\gamma = 32$ is more than adequate for nearly all purposes, and smaller values can be

¹That is, if all our knowledge about the data is contained in P then the expected code length across all realizations is bounded below by this expression, and the bound can be approached to better than 1 bit. The proof constructs a Huffman code directly from the probability distribution [9].

chosen if desired. Since it is common to work with natural logarithms in estimation theory, we shall do so from now on, and the code length will therefore be in nats rather than bits.

The total description length for a realization z of Z is (again ignoring the program for G)

$$L(z, \bar{\lambda}) = L(z|\bar{\lambda}) + L(\bar{\lambda}) \quad (2)$$

where the data code length

$$L(z|\bar{\lambda}) = -\ln P(z|\bar{\lambda})$$

is just the negative log likelihood of the data under the assumed distribution and the truncated parameter values. The MDL principle therefore requires us to minimize (2) over $\bar{\lambda}$.

The minimization could be done by trying all combinations of different numbers of bits for the different parameters, but this becomes infeasible very quickly as the number of parameters is increased. One could approximate by assuming all parameters are given to the same accuracy, or one could ignore the integer constraint on numbers of bits and treat δ as continuous. We take the latter approach here.

If the optimal δ_j values are not too great, $\bar{\lambda}$ will not be far from the maximum likelihood value $\hat{\lambda}$ which optimizes $L(z|\lambda)$ over λ , and

$$L(z|\bar{\lambda}) \leq L(z|\hat{\lambda}) + \frac{1}{2}\delta^\top Q\delta \quad (3)$$

where $Q = D_{\lambda\lambda}L(z|\hat{\lambda})$ is the second derivative matrix corresponding to the maximum likelihood solution. (Note that this is an inequality, since the actual precision of λ_j may be better than δ_j .) Using (3) in (2), we obtain

$$L(z, \bar{\lambda}) \leq L(z|\hat{\lambda}) + \frac{1}{2}\delta^\top Q\delta + k \ln \gamma - \sum_{j=1}^k \ln \delta_j \quad (4)$$

as an approximation to the total description length that is to be minimized. The right-hand side of (4) is a sum of two convex functions of δ and so has a unique minimum. Carrying out the minimization over δ gives

$$(Q\delta)_j = 1/\delta_j \quad (5)$$

for each j . This is easy to solve numerically, to give the parameter precision, say $\hat{\delta}$, which gives the optimal bound when λ is fixed at $\hat{\lambda}$. The bound on minimum description length is therefore, after substituting (5) into (4),

$$S_k(z) = L(z|\hat{\lambda}) + \left(\frac{1}{2} + \ln \gamma\right)k - \sum_{j=1}^k \ln \hat{\delta}_j. \quad (6)$$

Thus a good approximation to $\bar{\lambda}$ is the maximum likelihood value $\hat{\lambda}$, truncated according to the precision specified by the solution of (5), resulting in a description length bounded by $S_k(z)$. We shall minimize $S_k(z)$ to choose a good model; the only difference between this approximate version of MDL and the maximum likelihood method is that we have to account for the additional penalty term defining the truncation. But it is precisely this term which enables us to choose the appropriate size of model, since if we increase the number k of nonzero parameters, the negative log likelihood $L(z|\hat{\lambda})$ in (6) always decreases but the cost $k(\frac{1}{2} + \ln \gamma) - \sum_{j=1}^k \ln \hat{\delta}_j$ of the parameters generally increases.

The precision $\hat{\delta}_j$ has the useful property of being the minimum required precision of λ_j . This specification of parameters is perhaps more useful than stating standard deviations, because it gives an unequivocal range over which the parameters can vary and the model remain accurate. The usual custom of stating standard deviations is problematical because for nonlinear models the distribution of fitted parameters cannot be assumed to be Gaussian; nor is it guaranteed that the model will be accurate if all parameters are taken to the limits of their confidence intervals.

It is clear that the MDL criterion is related to other well-known model selection criteria, and Rissanen shows that asymptotically, our approximate expression for MDL is equivalent to the Schwarz (or Bayesian) information criterion [25, 11, 14, 22]. We have found, however, that working with the above form gives better results for smaller data sets, and for large ones in critical cases, and the extra computation required is not significant. Actually, we can do better still. The main reason for the approximation was to simplify the $\hat{\delta}$ calculation; it is straightforward to use the more accurate formula for L given in Appendix A.1 in the calculation of description length as a function of $\hat{\delta}$, and in fact we do this in the applications later.

An example is shown in Fig. 1, which shows the variation in description length as the number of parameters is increased in a model to be discussed later.

We remark that the MDL criterion could also be used to answer the perennial question of what is a good embedding for a given reconstruction problem, by using it to compare models with different embedding dimensions. The embedding dimension (and lag) come in as parameters to be specified and via the number of initial terms that must be specified in the time series to allow the first prediction to be made, as well as in the specification of model-dependent parameters such as location of centers in a radial basis model.

2 Pseudo-linear Models and Subset Selection

How should the subclass of nonlinear models $G(X, \lambda)$ be chosen? Perhaps the best criterion of choice is practical expediency. In the selection problem we are faced with estimating a large number of models and choosing the best among them. It would be advantageous to use a subclass of nonlinear model for which the fitting step is efficient. On the other hand, the subclass should model a wide variety of nonlinearities. For these reasons we prefer models which are linear combinations of nonlinear functions, that is, *pseudo-linear models*. Pseudo-linear models have the general form

$$G(x, \lambda) = \sum_{i=1}^m \lambda_i f_i(x) \tag{7}$$

for some arbitrary scalar functions f_i , called *basis functions*. The parameters λ_i of such a model are easily estimated. Unfortunately, if we use (7) as it stands, we only obtain a *weak* model, as discussed in Section 1.1. We will get round this problem, but defer the solution to the next section; let us first discuss the model selection problem.

The selection problem is to decide which of the components of $\lambda = (\lambda_1, \dots, \lambda_m)$ should be nonzero. A major purpose of this paper is to describe an algorithm that efficiently solves the selection problem for pseudo-linear models and to demonstrate the properties of these optimal models through example applications.

The main reason for using pseudo-linear models in parameter estimation is simply linear regression. Given a realization of process (1), one solves

$$\text{minimize } |y - V\lambda| \tag{8}$$

where $y = (y_1, \dots, y_n)^\top$, $\lambda = (\lambda_1, \dots, \lambda_m)$ and V is an $n \times m$ matrix whose i th column is

$$v_i = (f_i(x_1), \dots, f_i(x_n))^\top,$$

and $|\cdot|$ is a norm chosen to suit the assumed distribution of the errors ϵ_t . Typically one assumes errors are normally distributed, in which case the maximum likelihood estimate required by (6) is obtained by minimizing the L_2 -norm. Solving (8) for the parameters λ when using the L_2 -norm is a simple procedure using, say, singular value decomposition.

The linear models are trivial examples of pseudo-linear models, and many other familiar models fit into this general class: global polynomials, orthogonal functions, radial basis functions, single layer neural nets, piecewise linear models and piecewise polynomials, to mention a few. Many other models do not, especially those where parameters appear inside nonlinear functions, such as multi-layer neural nets and self-exciting threshold, auto-regressive (SETAR) models. A common trait of the alternatives to pseudo-linear models is they typically involve a computationally intensive parameter estimation process, such as non-linear regression or back-propagation. Pseudo-linear models avoid complex fitting procedures, but in their usual form they are *weak* models.

There are, of course, uncountably many possible basis functions and the selection problem is still intractable at this stage. However, the selection problem for pseudo-linear models can be solved in a restricted form by initially choosing a large number of basis functions f_i , which one hopes will capture every nonlinearity the time series might possess. In fact, it will be seen in the next section that with careful choice one can also ensure that *strong* models are generated. The selection problem is now to choose the smallest subset of basis functions that adequately models the time series: that is, select which components of λ should be nonzero. We will call this the *restricted selection problem*.

A practical algorithm that solves the restricted selection problem is as follows. For $k = 0, 1, \dots$,

$$\text{minimize } |y - V\lambda| \text{ subject to } \mathcal{N}(\lambda) = k \tag{9}$$

where $\mathcal{N}(\lambda)$ is the number of non-zero components of λ . Continue to increase k until there is no significant improvement in the model according to the MDL criterion. That is, we try to find the minimum of $S_k(z)$ over k by calculating it for successive values of k .

Calculating $S_k(z)$ involves a little manipulation: the formula is derived in Appendix A.2 and is (to within an irrelevant additive constant)

$$S_k(z) = \left(\frac{n}{2} - 1\right) \ln \frac{\hat{e}^\top \hat{e}}{n} + (k + 1) \left(\frac{1}{2} + \ln \gamma\right) - \sum_{j=1}^k \ln \hat{\delta}_j \tag{10}$$

Here $k = \mathcal{N}(\hat{\lambda})$, $\hat{e} = y - \hat{V}\hat{\lambda}$ is the error vector, \hat{V} is the n by k matrix formed from the columns of V corresponding to non-zero components of $\hat{\lambda}$, and $\hat{\delta}$ solves $Q\delta = 1/\delta$ as before, where

$$Q = \hat{V}^\top \hat{V} / \delta^2, \tag{11}$$

and $\hat{\sigma}^2 = \hat{e}^\top \hat{e}/n$ is the mean square fitting error. The expression for S_k was calculated on the basis of λ and σ^2 being the parameters to be found. We are suppressing the fact that the matrix V may—and for strong pseudo-linear models, will—contain parameters such as radial basis centers; this adds to the complexity of the approach and is left for a later paper.

3 Selection algorithm

Recall that the restricted selection problem involves solving (9) for a succession of values of k . The constrained minimization of (9) over λ is the nontrivial part.

Here we describe an efficient algorithm to solve the restricted selection problem for pseudo-linear models using quadratic programming techniques. It was derived from unpublished work we did on selection for robust pseudo-linear models but without minimum description length: in that case, we use the L_1 -norm and the selection model can be handled by a variant of the simplex algorithm. In the present case we apply similar methods to quadratic problems.

We note in passing that V might be constructed to contain only the columns required to build a linear model. The following algorithm still applies and will result in the construction of an optimal linear model. In Section 5.2 we observe that such a linear model may be superior to those constructed by standard procedures.

To enable efficient building of models of fixed size we employ a modified quadratic programming technique that uses sensitivity analysis. If the noise process is i.i.d. Gaussian, the maximum likelihood problem reduces to least squares fitting so the restricted selection problem involves choosing the optimal subset of size k at each stage, so as to minimize $(y - V\lambda)^\top (y - V\lambda)$. It is convenient to write the problem as

$$\begin{aligned} & \text{minimize} && \frac{1}{2}e^\top e \\ & \text{over} && e, \lambda \\ & \text{subject to} && V\lambda + e = y, \quad \mathcal{N}(\lambda) = k. \end{aligned} \tag{12}$$

The difficult part here is the constraint $\mathcal{N}(\lambda) = k$. Let B be the set of *basic* indices, namely

$$B = \{j : \lambda_j \neq 0\},$$

so $\mathcal{N}(\lambda) = |B|$. We are going to use sensitivity analysis to see the effect of changing the size of B . The constraint $\mathcal{N}(\lambda) = k$ becomes

$$\lambda_j = u_j, \quad j \notin B, \tag{13}$$

where $u = 0$ but is kept as a parameter. It is an easy exercise in Lagrangian theory [26] to discover the sensitivity of the optimal solution $\psi(y, u)$ of (12) to changes in u .

First we consider how to enlarge the set of basic variables so as to give the greatest benefit to the mean square error. The Lagrangian of the problem (12) after replacing $\mathcal{N}(\lambda) = k$ with (13) is

$$\frac{1}{2}e^\top e + \nu^\top (y - V\lambda - e) + \mu^\top (u - \lambda)$$

where μ and ν are dual variables. Optimizing over e and λ without constraints gives $\nu = e$ and

$$V^\top \nu + \mu = 0.$$

Thus

$$\mu = -V^\top e$$

is the dual variable corresponding to constraint (13), and so is the sensitivity to changes in u at optimality. That is,

$$\mu_j = \frac{\partial \psi}{\partial u_j}.$$

The largest element of μ in absolute value tells us which index should be made basic to give the greatest marginal improvement in payoff. This will not necessarily be the one which gives the greatest actual improvement, but we hope to overcome this problem by using an iterative scheme.

Now we consider how to shrink the set of basic variables so as to do the least damage to the mean square error. The dual optimization problem is obtained by substituting the optimal values of λ and e into the Lagrangian, and is

$$\begin{aligned} & \text{maximize} && -\frac{1}{2}\nu^\top \nu + \nu^\top y + \mu^\top u \\ & \text{over} && \mu, \nu \\ & \text{subject to} && (V^\top \nu)_j = w_j \quad \text{for } j \in B \end{aligned} \tag{14}$$

where $w = 0$, but is kept as a parameter, and we substitute $u = 0$ at once. Notice that the constraint involving w now affects the basic variables so the sensitivity to w tells us about variables already in the basis. The problem is convex so the primal and dual solutions are the same, and it is not hard to check that the dual variable to μ is λ , so that

$$\lambda_j = \frac{\partial \psi}{\partial w_j}.$$

The smallest element of λ_j in absolute value tells us which variable to make non-basic to do the least marginal damage to the payoff.

The above two observations provide a simple means to find a good model with k basis vectors by adding, then deleting, basis vectors until the basis no longer changes. Hence, if one assumes that the optimal model with k parameters is close to the optimal model with $k + 1$ parameters, then the following algorithm efficiently solves (12).

Let V_B be the $n \times k$ matrix formed from the columns of V with indices in B , let λ_B be the least squares solution to $y = V_B \lambda$, and let $e_B = y - V_B \lambda_B$.

1. Let $S_0 = (\frac{n}{2} - 1) \ln(y^\top y/n) + \frac{1}{2} + \ln \gamma$.
2. Let $B = \{j\}$ where V_j is the column of V such that $|V_j^\top y|$ is maximum. (Note that $\lambda_B = V_j^\top y / V_j^\top V_j$ in this case.)
3. Let $\mu = V^\top e_B$ and i be the index of the component of μ with maximum absolute value. (Index i is the index coming *in* to the basis.) Let $B' = B \cup \{i\}$.
4. Calculate $\lambda_{B'}$. Let o be the index in B' corresponding to the component of $\lambda_{B'}$ with smallest absolute value. (Index o is the index going *out* of the basis.)
5. If $i \neq o$, then put $B = B' \setminus \{o\}$ and go to step 3.
6. Define $B_k = B$, where $k = |B|$. Find δ such that $(V_B^\top V_B \delta)_j = 1/\delta_j$ for each $j = \{1, \dots, k\}$ and calculate $S_k = (\frac{n}{2} - 1) \ln \frac{\hat{e}^\top \hat{e}}{n} + (k + 1)(\frac{1}{2} + \ln \gamma) - \sum_{j=1}^k \ln \hat{\delta}_j$.

7. If $S_k < S_{k-1}$ (see note on stopping criterion below), then go to step 3.
8. Take the basis B_k such that S_k is minimum as the optimal model.

3.1 Stopping criterion

Sometimes, the behavior of the selected value of S_k as a function of k is not quite as simple as one might hope. Usually, S_k decreases rapidly as k is increased, then slowly increases. Around the minimum the value of S_k usually fluctuates, probably because we are failing to find the global minimum at every value of k . To allow for this fluctuation it is necessary to weaken the stopping criterion of the selection algorithms. A suitable weakening is not to stop immediately $S_k < S_{k+1}$, but to wait until $S_k < S_{k+l}$ for $l = 1, \dots, p$ (the number p being arbitrary, with a value around 10 being good in our experience).

3.2 Some compromises in our approach

It is really necessary to allow for the fact that ϵ_t are correlated and have a distribution that depends on X_t . Our algorithm simply ignores this subtlety, but we have found that a test of normality of the residues often fails due to excessive numbers of outliers. One can choose to ignore this too, or try more sophisticated methods [16], or take a robust statistics approach and use an L_1 -norm. The latter alternative requires some additional development in applying the MDL criterion.

As we pointed out earlier, we have not included the accuracy of the centers in our measure of description length. This means our models are probably somewhat over-fitted, but in our experience it is unlikely that this would change any of our conclusions. We decided that to include the rather messy algebraic details of how to treat centers as parameters in the MDL calculation would have decreased the comprehensibility without greatly adding to the usefulness of the approach. In a later paper we hope to go into fuller detail on this.

3.3 Comparison of model selection algorithms

Another contender as a model selection method is the orthogonal least squares method [4, 14], which is the following iterative procedure.

1. Let $k = 0$, $V_0 = V$ and $e_0 = y$.
2. At the k th step find the column v_{i_k} of V_k that maximizes $e_k^\top v_i / |v_i|$. That is, find the column of V_k that gives the best least squares fit to the “error” vector e_k .
3. Now find that part of e_k and the columns of V_k that are orthogonal to v_j for $j = i_1, i_2, \dots, i_k$ (i.e. perform a Gram-Schmidt reduction) and call these e_{k+1} and V_{k+1} .
4. Stop if some suitable criterion is satisfied; otherwise, increment k and return to step 2.

The models selected are formed from the k vectors $v_{i_1}, v_{i_2}, \dots, v_{i_k}$ that have been selected. The MDL criterion can be used to assess when to stop selecting vectors.

This method builds a best least squares model of y from the columns of V so as to choose at each step the column that best accounts for the error of the current model. Note that once a column enters a model it is used in all larger models; compare this with the above algorithm which can remove a column from the model at any time and replace it with a better choice.

In our experiments, the algorithm described above usually does better than the orthogonal least squares method in terms of both mean square error and MDL criterion.

Other common techniques that resemble our algorithm are stepwise regression and singular value decomposition; the resemblance is, however, only superficial. Successful stepwise regression requires an a priori selection of a small set of basis functions that have as little correlation as possible. Since our algorithm essentially selects a linearly independent basis from V in which y has “special” projection properties, the problems of stepwise regression are avoided. Singular value decomposition, on the other hand, describes y in terms of the principal components of V , and often tells us which principal components are important. However, the principal components tell us little about which basis functions are important.

3.4 Are selected subsets globally optimal?

Notice that the algorithm obtains an “optimal” basis of size $k + 1$ from one of size k by swapping a basic vector with a non-basic vector to improve on the current basis. This implicitly assumes the optimal model of k parameters is close to the optimal model of $k + 1$ parameters. The assumption is not necessarily correct and the following is an example of a system where it is not true.

$$V = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & -2 & -3 \end{pmatrix}, \quad y = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \quad (15)$$

The optimal model of 1 parameter is either column 1 or 2, and the optimal model of 2 parameters is columns 3 and 4. Having introduced column 1 or 2 into the basis, the optimal basis obtained by introducing only one more vector is column 1 and 2. It is impossible to arrive at the optimal 2-parameter solution without simultaneously swapping *two* vectors or temporarily making the basis larger. Modifying the algorithm to swap more than one vector at a time will overcome this difficulty at the expense of increasing computational effort. However, example (15) can be generalized to a system where for any m , swapping fewer than m vectors simultaneously will fail to find the optimal model.

The orthogonal least squares method, incidentally, fails in even simpler cases.

The difficulty illustrated by (15) is not an isolated example. On the contrary, it appears that finding the optimal model of size k is NP-hard—related to the feasible basis extension problem [18, 5]—although we have not established this rigorously. If this is the case, then we cannot expect to obtain the optimal solution easily. On the other hand, numerical tests with real time series suggest that this simple algorithm compares very favorably, in terms of mean square error and MDL criterion, with more exhaustive algorithms. Also, the counter-example is contrived and the matrix used has a structure quite unlike the structure of a matrix V one would expect from an embedded time series. Finally, the solution to the restricted selection problem is only an approximation to a solution to the selection problem, which is in its turn only an approximation to the optimal non-linear model. Consequently, a good approximate solution to the restricted selection problem, which this algorithm provides, ought to be a good approximate solution to the optimal nonlinear model and the marginal gain from more exhaustive algorithms is unlikely to be worth the effort.

4 Choosing basis functions to generate strong models

Pseudo-linear models are *weak* models and it is our preference to obtain *strong* models. In this section we describe some methods that will assist in generating a strong model. To obtain a strong model it is necessary, though not sufficient, to have some nonlinear parameter dependence [3]. For example, a conventional feed-forward neural network with one hidden layer has $f_i(x) = \phi(w_i x - \xi_i)$ in (7) resulting in nonlinear dependence on its input weights w_i and thresholds ξ_i ; a radial basis function model has $f_i(x) = \phi(|x - c_i|)$ resulting in nonlinear dependence on the location of its centers c_i .

Several methods exist for adjusting nonlinear parameters. A common technique begins with k basis functions with arbitrarily chosen parameters, then adjusts the parameters by gradient descent to find an optimal model. Another technique makes a grid search over a region of parameter space. However, we observe, in the light of Section 1.2, that parameters need only be specified to some precision. If one is lucky, or careful, the precision required of nonlinear parameters is much less than that required for λ parameters, and hence accurate adjustment of nonlinear parameters may not be critical to a model's performance. This does seem to be supported by experience in using radial basis functions, where the literature is full of good models built from even randomly chosen centers.

Consequently, we propose the following method to optimize the nonlinear parameters: initially choose a large number of basis functions with various arbitrary values for the nonlinear parameters, and then select the k basis functions that give the best model. Of course, this is what we have already called the restricted-selection problem; now, however, we are using the fact that if enough basis functions were initially chosen, at least some of them would lie near to the optimal values for the nonlinear parameters and be indistinguishable at the precision required of the optimal values.

We also remark that if we are going to generate a large number of basis functions, they need not all be of the same form: it is often advantageous to mix constant and linear functions with radial basis functions of different types (and scales, for radial basis functions that require a scale parameter). Including a constant function and projections onto the coordinate axes among the candidate functions (so V contains a column of ones and copies of all the columns of x_1, \dots, x_n)^T) is appropriate because one should always test that a linear model is not sufficient before considering a nonlinear model. Furthermore, it might be that a linear model accounts for a substantial amount of the dynamics even when it is not actually optimal. An important point to appreciate is that if the constant and linear functions are included then the selection algorithm of Section 3 automatically tests whether a linear model is sufficient because if it is, then the only non-zero components of the optimal λ_k will correspond to constant and linear basis functions.

To some extent, the nonlinear basis functions that are initially chosen will depend on the particular nonlinearities one anticipates in the time-series. Generally speaking, a poor choice of basis functions will result in a model with more parameters than one built with a better choice, but usually not in an inferior model in terms of fit—more precisely, a poorer description length but comparable mean square error.

4.1 Radial basis models

Radial basis models are an appealing class of pseudo-linear model; they have been advocated by many authors and we use them in our numerical examples and make some specific ob-

servations regarding them in the remainder of this Section. Our methods are not restricted to use with radial basis models and could be modified appropriately to feed-forward neural networks and to other approximation schemes.

There has been much discussion on how to select ϕ and common choices include the cubic, the thin-plate spline function, and the Gaussian. Radial basis models are strong models because the centers (and scale radii where appropriate) should also be considered as parameters and it is these parameters that provide the essential nonlinear dependence of strong models. As we have seen, choosing the centers becomes the crux of the model selection problem. A number of methods have been proposed: using data points or randomly selected subsets of these, randomly selected points in a bounding box, and local centroids or k -means. Although we have our favorite methods, in this paper we will simply use a variant of random center selection.

Choosing centers randomly in a bounding box does not work well because typically data lives in a fairly confined region which is a small subset of the box, and most centers are too far away to be useful. Using data points is a better method. If there are too many data points then a randomly selected subset of these is acceptable; the latter, in effect, is choosing centers by the “natural” measure of a supposed attractor. The method of k -means [12, 15] is more sophisticated; it tries to divide the data points into k -clusters and uses the centroids of these clusters as the centers.

A common property of all but the first method is that the centers chosen always lie within the convex hull of the data; indeed they lie within a region less than the convex hull. We argue, and have found by experiment, that it is often advantageous to use centers somewhat outside the region occupied by the data, while not wasting parameters by selecting them over a bounding box. We will call such centers near but not on the data region *chaperons*.

A straightforward and effective method to generate chaperons is to add noise to the data; enough noise should be added to put some chaperons outside the region occupied by the data, but not too far from it: using Gaussian noise with say, 30% of the standard deviation of the data seems to be adequate. Note that chaperons also enable us to generate a large number of distinct basis functions by taking several noisy copies of each datum, which is convenient and effective regardless of whether the data set is large or small.

Now we simply use the selection algorithm of Section 3 to choose those columns of V which give the best fit to the data for any given number of columns. It is possible to improve the fit by fine-tuning the locations of the selected centers using a nonlinear optimization algorithm, but the improvement in fit is not generally very great.

In this paper we will keep things simple: we merely select a large set of chaperons and apply the selection algorithm without further refinement. With such a crude approach, it is worthwhile running the algorithm a few times, with different sets of chaperons, and using description length to select the best.

Fig. 1 earlier in this paper showed the behavior of the description length when this approach was used to build models from sunspot data; the problem will be discussed in more detail later

5 Applications

We will consider two distinct applications of optimal strong pseudo-linear models: as a means to make statements about a dynamical system that generates a time series and as a means

to predict the future of a time series. The latter is a classical application and we shall make some comparisons of the predictions of linear models, pseudo-linear models and some other nonlinear models.

5.1 Detecting Shil'nikov bifurcations from data

Our first application stems from a theorem of Takens which asserts that if a times series generated by dynamical system is embedded suitably, then one obtains a system that retains much of the original system's local and global properties [24, 19]. Since an optimal model \hat{F} should only extract the dynamical, deterministic component of a time series, it might also have implicit in it properties of the original system that are invariant under Takens' transformation. We might then analyze the dynamical system defined by \hat{F} as we would a map derived from theoretical analysis by, for example, finding fixed points, eigenvalues and homoclinic orbits [8]. There are important technical issues that need to be addressed regarding just what remains invariant under a Takens transformation, but these are outside the scope of this paper. The aim here is to demonstrate that above scheme is feasible.

A frequently asked question is, given a time series, can it be determined whether the generating system is chaotic or not? In the presence of measurement error this question cannot be answered by estimation of simple statistics such as fractal dimension. However, if one can show, for example, that the conditions of Shil'nikov's bifurcation theorem are satisfied for a homoclinic orbit of \hat{F} , then the existence of chaos is better established. We demonstrate with an artificial system that a Shil'nikov bifurcation can be recognized from data alone and refer the reader to a future report by the authors where similar conclusions are drawn from experimental data.

First we give a brief description of the Shil'nikov bifurcation; for details the reader should refer elsewhere [7, 17]. Suppose a system of differential equations $\dot{x} = f(x, \mu)$, $x \in \mathbf{R}^3$, $\mu \in \mathbf{R}$, with f analytic in x and μ , has a fixed point \hat{x}_μ with one real eigenvalue $\lambda_\mu > 0$ and a complex conjugate pair $\sigma_\mu \pm i\omega_\mu$, and that \hat{x}_μ has a homoclinic orbit when $\mu = 0$ but not necessarily otherwise. Define $\delta = |\sigma_0|/\lambda_0$. The theorem of Shil'nikov describes the behavior of this system in the neighborhood of the homoclinic orbit for given δ . The naive interpretation of this theorem is, if the system is close to having a homoclinic orbit, that is, μ is close to zero, then one might expect to see a stable periodic orbit for $\delta > 1$ and chaotic motions for $\delta < 1$. The theorem does not guarantee these observations, but they would be consistent with the invariant structures the theorem implies. See the above references for an accurate statement of the theorem and discussion of the global dynamics.

Consider the following dynamical system,

$$\dot{x} = -\sigma x - \omega y, \tag{16}$$

$$\dot{y} = \omega x - \sigma y + z^2, \tag{17}$$

$$\dot{z} = \mu + \lambda z - x^2 z, \quad \sigma, \lambda > 0. \tag{18}$$

This system has three fixed points when $\mu \approx 0$. One of these fixed points is close to the origin with eigenvalues close to λ and $-\sigma \pm i\omega$ and corresponding unstable and stable eigenspaces nearly parallel to the z -axis and xy -plane. By numerical integration and continuity arguments it is easily established that a homoclinic orbit exists for some $\mu \approx 0$. Hence, a Shil'nikov bifurcation [7] must occur in the neighborhood of $\mu = 0$ and for small values of μ we expect chaotic behavior for $\sigma/\lambda < 1$. Fig. 2 shows a trajectory of the system in a likely chaotic region $\sigma = 1$, $\omega = 15$, $\mu = 0.01$ and $\lambda = 6$.

We now ask, if given a time series, such as the $x + z$ -values of the trajectory shown in Fig. 2, possibly corrupted by noise, is it possible to infer a Shil'nikov mechanism? In the rest of this section we demonstrate how this can be done, omitting some technical details.

We use a simple time-delay embedding $X_t = (Y_{t-\tau}, Y_{t-2\tau}, \dots, Y_{t-d\tau})$. The process (1) defined by F defines a discrete dynamical system on embedding space,

$$X_t \mapsto (F(X_t), Y_{t-\tau}, \dots, Y_{t-(d-1)\tau}). \quad (19)$$

If the time series Y_t is generated by a dynamical system and $\epsilon_t = 0$ for all t , then Takens' theorem [24, 19] states that for d sufficiently large (19) has equivalent dynamics to the generating system integrated over some time step T . (One of the technical issues we avoid here is whether the dynamics of the embedded system has additional features that might mislead us; in brief, we do not expect to be misled as long as we only look at orbits that remain close to the original embedded data points.) Given a time series we can approximate this dynamical system using an optimal model \hat{F} in place of F . Our aim here is to analyze the dynamical system defined by \hat{F} , assuming that in the neighborhood of the fitted data something of the local and global properties of the original system are preserved.

We take rather a short time series, of 400 $x + z$ -values sampled every $T = 0.02$ time units, to which we add white noise at a signal-to-noise ratio of 2%. We choose the lag to be $\tau = 5$ which is a little over one quarter of the average inter-peak period of the time series [2]. The method of false nearest neighbors [1] suggests that for this lag a three dimensional embedding is sufficient. We obtained an optimal model \hat{F} for F for a prediction of τ steps ahead using a radial basis model formed from Gaussian radial basis functions, $\phi(r) = \exp(-r^2/2\sigma^2)$, where σ^2 is the variance of the time series, and chaperons as centers generated by 30% noise on the embedded data and incorporating linear basis functions.

The dynamical system defined by a time-delay embedding model can be written

$$(x, y, z) \mapsto (\hat{F}(x, y, z), x, y). \quad (20)$$

The fixed points of (20) are $x = y = z$ and $x = \hat{F}(x, x, x)$. Furthermore, the eigenvalues and eigenspaces of the linearized system at a fixed point are given by the Jacobian

$$\begin{pmatrix} \partial\hat{F}/\partial x & \partial\hat{F}/\partial y & \partial\hat{F}/\partial z \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad (21)$$

where it is noted that for pseudo-linear models the partial derivatives are easily calculated, and exist provided the basis functions are differentiable. The discrete dynamical system (20) is supposed to result from integrating a flow over a time period T . Fixed points of the map correspond to fixed points of the flow, and likewise there is a correspondence of invariant eigenspaces. The value of T should be chosen small enough that the flow in small neighborhoods of fixed points is nearly linear over a time T . With such T the relationship between the eigenvalue λ of an eigenspace of the flow and the eigenvalue μ of the corresponding eigenspace of the map is such that $|\mu|$ is proportional to $e^{T \operatorname{Re}\lambda}$.

Table 1 lists the fixed points and eigenvalues of the system for the chosen value of T and Table 2 lists those of our radial basis model. The locations of the estimated fixed points and their eigenspaces are shown in Fig. 3. We have found the correct number of fixed points, and their locations are reasonable. The logarithms of the absolute values of the eigenvalues of the

x	y	z	α	β	δ
-2.45	0.16	-6.07	-13.88	$5.94 \pm 18.86i$	0.43
0.00	0.00	0.00	6.00	$-1.00 \pm 15.00i$	0.17
-2.45	0.16	6.08	-13.88	$5.94 \pm 18.86i$	0.43

Table 1: Fixed points and eigenvalues for the system (20). The eigenvalues are α and β , and $\delta = -\text{Re}\beta/|\alpha|$.

fixed points should be proportional to the real parts of the eigenvalues in the corresponding flow, but the constant of proportionality depends on details of the embedding. However, the Shil'nikov theorem only requires calculation of their ratio δ , from which this constant cancels. The values of δ are listed in the tables and it is seen that although the estimated values are poor, they are all still less than 1; similarly, the signatures of the eigenvalues are correct.

Fixed Point	Location	α	β	δ
A	-3.68	0.56	$0.04 \pm 1.64i$	0.86
B	-0.42	1.29	$0.01 \pm 0.82i$	0.75
C	2.41	-0.03	$0.09 \pm 1.03i$	0.008

Table 2: Fixed points and eigenvalues for radial basis model of a Shil'nikov bifurcation. The locations of the fixed points are (x, x, x) because a simple time-delay embedding is used. The eigenvalues are α and β , and $\delta = -\log|\beta|/|\log\alpha|$.

In Fig. 3 it can be seen that fixed point B is implicated as the source of the Shil'nikov bifurcation and it is only this δ that is significant. Closer examination of Fig. 3 reveals that the properties of fixed points A and C are deduced almost entirely from the global structure of the flow, whereas for B there is a comparatively large amount of local information; consequently, it would not be surprising if even the signatures of the eigenvalues of A and C were wrong. In all three cases, the orientations of stable and unstable eigenspaces are approximately correct.

In order to finally infer a Shil'nikov bifurcation, it remains to be shown that the dynamical system has, or is close to having, a suitable homoclinic orbit. In the map such an orbit should correspond to the unstable manifold of the fixed point just identified. The approximate location of the unstable manifold can be traced by iterating a point close to the fixed point B and in its unstable eigenspace. We observe that such a trajectory quickly moves into the region of the stable eigenspace of the fixed point and spirals in toward the fixed point in a way one should expect in a Shil'nikov bifurcation.

Hence, we can infer from this time series alone that the apparently chaotic motions are likely to be the result of a Shil'nikov bifurcation.

5.2 Time series prediction

The quality of a prediction is usually measured by the mean square error of a series of predictions, which reflects an assumption that prediction errors have a normal distribution. There are two common forms of prediction which we shall call *one-step* predictions and *free-run* predictions. In a one-step prediction Y_t is calculated given Y_s , $s < t$ and when Y_t becomes known, then it is used to predict Y_{t+1} and so on. In free-run predictions, sometimes called multi-step predictions, $Y_{t+\tau}$, $\tau \geq 0$ are predicted given Y_s , $s < t$. Free-run predictions are the most demanding of a model. We will consider free-run prediction of the annual sunspot numbers time series, which has received considerable attention and is recognized as being a difficult prediction problem.

Ghaddar and Tong have considered the problem of building a model using annual sunspot numbers over the period 1700–1979, then using this model to make free-run predictions for the period 1980–1987 [6, 25]. Tong compares the predictions of optimal auto-regressive model AR(9) and an optimal threshold auto-regressive model SETAR(2;3,11) [25], which is also a pseudo-linear model. To facilitate comparison with Tong’s results, the SETAR model was constructed from instantaneously transformed data, $y_t = 2\sqrt{s_t + 1} - 1$, where s_t is the raw annual sunspot average. Both models used Akaike’s criterion (AIC) as the model selection criterion. The optimal auto-regressive model was found to be

$$\begin{aligned} Y_t = & 6.9627 + 1.2064Y_{t-1} - 0.4507Y_{t-2} - 0.1747Y_{t-3} + 0.1974Y_{t-4} \\ & - 0.1366Y_{t-5} + 0.0268Y_{t-6} + 0.0128Y_{t-7} - 0.0312Y_{t-8} \\ & + 0.2123Y_{t-9} + \epsilon_t, \end{aligned} \quad (22)$$

with a mean sum of squares of residuals of 221.

As mentioned previously, our model selection algorithm can be applied to purely linear models. Using our algorithm to build the optimal auto-regressive model with lags up to 9, we obtained

$$Y_t = 5.1968 + 1.2221Y_{t-1} - 0.5229Y_{t-2} + 0.2070Y_{t-9} + \epsilon_t, \quad (23)$$

with a mean sum of squares of residuals of 225. The same model is obtained using any of the AIC, BIC (Bayesian) or MDL selection criteria. We note that this model has insignificantly worse mean square error, but significantly fewer parameters. The MDL criterion asserts that the required precisions of the parameters are 0.0322, 0.0008, 0.0005 and 0.0006 in respective order of appearance in (23). This example demonstrates that our model selection algorithms can provide superior linear models compared to the naive approach of simply progressively increasing the model order.

Table 3 shows the computed mean square error for free-run predictions of the 1980–1987 annual average sunspot numbers using the AR(9) model (22), the reduced auto-regressive model (23), Tong’s SETAR(2;3,11) [25] and a radial basis model constructed on the instantaneously transformed data y_t using an embedding dimension of 6 and lag of 2, radial basis function $\phi(r) = \exp(-r^2/2\sigma^2)$, where σ^2 is the sample variance of the time series $\{y_t\}$, and with centers taken as 1000 chaperons generated by adding 30% noise to uniform random selections from the embedded data. Fig. 1 earlier in this paper showed the variation of description length with the number of selected parameters for this model.

It is clear from Table 3 that over the 1980–1987 period the SETAR model out-performs all other predictors and the reduced auto-regressive model (23) is a distant second. However, Table 3 also shows that if the prediction period is extended to include 1988, then there is a

significant reordering of success. It is probably unwise to attach too much to this observation, but it is worth noting why one extra prediction should make such a significant difference. Fig. 4 shows the annual average sunspots numbers and predictions of the models. It is well known that annual average sunspot numbers *increase* more rapidly than they *decrease*, which is in itself an indication of the nonlinearity of this time series. During the period 1980–1987 the sunspots numbers were in a decline, followed by a dramatic increase over 1987–1988. Of the models considered here the one that best captures this particular rapid increase, and the general asymmetry of rise and fall, is the radial basis model. The linear models will never capture the asymmetry of rise and fall, since their periodic behavior is always symmetric, and over longer prediction periods they perform progressively worse. The SETAR model is very good at modeling the *fall* in sunspot numbers, hence its success over the 1980–1987 period.

Model	Mean sum of squares of residuals	
	1980–1987	1980–1988
AR(9)	190	334
SETAR	55.8	413
reduced AR	130	214
Radial basis	288	306

Table 3: Errors for free-run predictions of sunspot numbers in 1980–1987 and 1980–1988, using various models built from data from earlier years.

It is worth remarking that when the radial basis model free-runs for a long time, it settles to a limit cycle. The limit cycle is not very helpful for prediction since the original time series has non-constant “period” and so there is phase slippage between the prediction and the reality. The fact that the best nonlinear model according to our criterion is periodic may be disappointing to anyone hoping for a chaotic model of the sunspot cycle, and we remind readers that we can only claim to have found (nearly) the best model within the classes we have tried: we cannot definitively state that no model exists with better prediction properties. However, it is something of an achievement to achieve even periodicity rather than divergence, or convergence to an equilibrium, which are commonly seen with nonlinear reconstructions. Moreover, we suggest that this may in fact be correct.

That is, we suggest the low-dimensional part of the sunspot cycle may well be periodic, the deviations from periodicity being due to high-dimensional or random effects. As evidence, consider Fig. 5 which shows a free-run with dynamic noise. This was obtained by using the random process (1) with \hat{F} substituted for F and Gaussian noise added. The result looks very like a real sunspot series, with apparent intervals of low and high amplitude and variable periodicity. It is easy to envisage the geometry of how dynamic noise can have this effect on a periodic attractor.

ACKNOWLEDGMENTS

This research was supported by a grant from the Australian Research Council. AIM thanks Kazu Aihara and Giles Auchmuty for interesting conversations and thanks Toyota Motor

Corporation and the University of Tokyo for support.

A Appendix

A.1 Description length of parameters

Suppose λ_j is expressed as the normalized floating-point binary number $0.1a_2a_3\dots \times 2^{m_j}$ where $a_i \in \{0, 1\}$. The convention is to choose m_j so that the first digit in the fractional part is 1; that digit is therefore redundant and can be used to represent the sign of λ_j . (The reader is invited to consider what to do with the value $\lambda_j = 0$.)

If λ_j is now truncated to $\bar{\lambda}_j = 0.1a_2a_3\dots a_{n_j} \times 2^{m_j}$, then the error is at most

$$\delta_j = 2^{-n_j}.$$

That is, we specify the j th parameter to n_j bits accuracy. To encode λ_j we have to encode the fractional part, which already includes the sign, and the value and sign of the exponent. That is, we have to encode a positive integer and a signed integer.

To encode integers we use a method described by Rissanen [20]. An arbitrary positive integer p can be described in $\log p$ bits (where for the moment, logarithms are to base 2), but we have to indicate somehow where the codeword ends. A good way is to present the codeword's length before the codeword: that is, to send the value of $\log p$ first. This requires $\log \log p$ bits, which is an integer that needs $\log \log \log p$ bits to encode, and so on; we can stop as soon as the number of bits required is 1 or fewer, as long as we have a special code reserved for small integers. Thus the code length is

$$L^*(p) = \log c + \log p + \log \log p + \log \log \log p + \dots$$

where the series terminates as soon as one of the terms is reduced to 0 or 1 and c represents the overhead to deal with small integers. Rissanen shows that a decipherable code can be defined in this way and that the shortest code length corresponds to $c \approx 2.865$. This in turn implies a probability distribution on the integers, which he calls the universal prior, and the code is optimal with respect to this prior.

Returning to our problem, we have to encode the fractional part, the exponent, and the sign of the exponent. Because it is traditional to work with natural logarithms in estimation theory, we shall work with $-\ln \delta_j$ rather than n_j ; as mentioned earlier, the lengths will be in nats rather than bits. Assume that L^* is defined to return nats rather than bits: the cost of all parameters in nats is therefore

$$L(\bar{\lambda}) = \sum_{j=1}^k L^*(\lceil 1/\delta_j \rceil) + \sum_{j=1}^k L^*(\lceil \ln(2 \max\{\bar{\lambda}_j, 1/\bar{\lambda}_j\}) \rceil), \quad (24)$$

where $\lceil x \rceil$ is the smallest integer not exceeding x and the terms describe the costs of fractional parts and exponents respectively; the factor of 2 represents the effect of the sign bit.

Although all of the terms in (24) are required for an optimal coding, the terms $-\sum \ln \delta_j$ dominate because the $\log \log \dots$ terms are slowly-varying. The exponent code can be simplified if we assume that the parameters take values in some fixed range (say, 10^{-9} to 10^9 ,

corresponding to about 2^{-30} to 2^{30}); then the exponent cost is fixed at, say, 5 bits. If we replace the right side of (24) by these dominant terms we now have

$$\tilde{L}(\bar{\lambda}) = \sum_{j=1}^k \ln \frac{\gamma}{\delta_j} \quad (25)$$

as a good approximation to L , where γ would be 32 in the above example.

A great advantage of this approximate formula is that the function \tilde{L} is convex as a function of δ and has continuous derivative, which enormously simplifies the optimization in Section 1.2. (Actually, the description lengths quoted in the examples in Section 5 use the correct formula (24) but with δ values calculated from (25). The value of γ is therefore irrelevant since it does not enter the calculation for $\hat{\delta}$.)

A.2 Minimum description length for pseudo-linear models

In this appendix we derive the formula for minimum description length of the pseudo-linear model used in section 2. We start with equation (6), repeated here as

$$S_k(z) = L(z|\hat{\lambda}) + k\left(\frac{1}{2} + \ln \gamma\right) - \sum_{j=1}^k \ln \hat{\delta}_j. \quad (26)$$

Here k is the number of parameters estimated by maximum likelihood, L is the negative log likelihood, and $\hat{\delta}$ is obtained by solving equation (5), which in turn requires finding $Q = D_{\lambda\lambda}L(z|\hat{\lambda})$, the second derivative matrix at the maximum likelihood solution.

In the pseudo-linear model, the errors are assumed independent Gaussians with mean 0 and unknown variance σ^2 . Thus the parameters are λ_j , $j = 1, \dots, k$ and also σ^2 so that (keeping k as the dimension of λ) we must replace (26) by

$$S_k(z) = L(z|\hat{\lambda}, \hat{\sigma}^2) + (k+1)\left(\frac{1}{2} + \ln \gamma\right) - \sum_{j=1}^k \ln \hat{\delta}_j - \ln \hat{\eta} \quad (27)$$

where η is the truncation bound on σ^2 , and the maximum likelihood values of λ and σ^2 are to be substituted into L . Here, $\hat{\delta}$ and $\hat{\eta}$ are the solutions of

$$Q \begin{pmatrix} \delta \\ \eta \end{pmatrix} = \begin{pmatrix} 1/\delta \\ 1/\eta \end{pmatrix} \quad (28)$$

where

$$Q = \begin{pmatrix} \partial^2 L / \partial \lambda^2 & \partial^2 L / \partial \lambda \partial \sigma^2 \\ \partial^2 L / \partial \sigma^2 \partial \lambda & \partial^2 L / \partial (\sigma^2)^2 \end{pmatrix}$$

is to be evaluated at the maximum likelihood values of λ and σ^2 .

By differentiating

$$L(z|\lambda, \sigma^2) = -\ln \left(\frac{1}{(2\pi\sigma^2)^{n/2}} e^{-(y-V\lambda)^\top (y-V\lambda)/2\sigma^2} \right)$$

with respect to λ and σ^2 we obtain the usual equations

$$V^\top (y - V\hat{\lambda}) = 0$$

and

$$\hat{\sigma}^2 = (y - V\hat{\lambda})^\top (y - V\hat{\lambda})/n$$

defining the maximum likelihood values.

Differentiating again we obtain

$$Q = \begin{pmatrix} V^\top V/\hat{\sigma}^2 & 0 \\ 0 & n/2\hat{\sigma}^4 \end{pmatrix}$$

where the cross-derivatives vanish, so we can solve (28) separately for $\hat{\eta}$ and $\hat{\delta}$. The solution for $\hat{\eta}$ is therefore

$$\hat{\eta} = \hat{\sigma}^2 \sqrt{\frac{2}{n}}$$

as one might perhaps expect.

Substituting for $\hat{\lambda}$, $\hat{\sigma}^2$ and $\hat{\eta}$ in (27) and writing $\hat{e} = y - V\hat{\lambda}$ we obtain finally

$$S_k(z) = \left(\frac{n}{2} - 1\right) \ln \frac{\hat{e}^\top \hat{e}}{n} + (k+1) \left(\frac{1}{2} + \ln \gamma\right) - \sum_{j=1}^k \ln \hat{\delta}_j + C \quad (29)$$

where C is independent of the parameters and in fact is given by

$$C = \frac{n}{2}(1 + \ln 2\pi) + \frac{1}{2} \ln \frac{n}{2}.$$

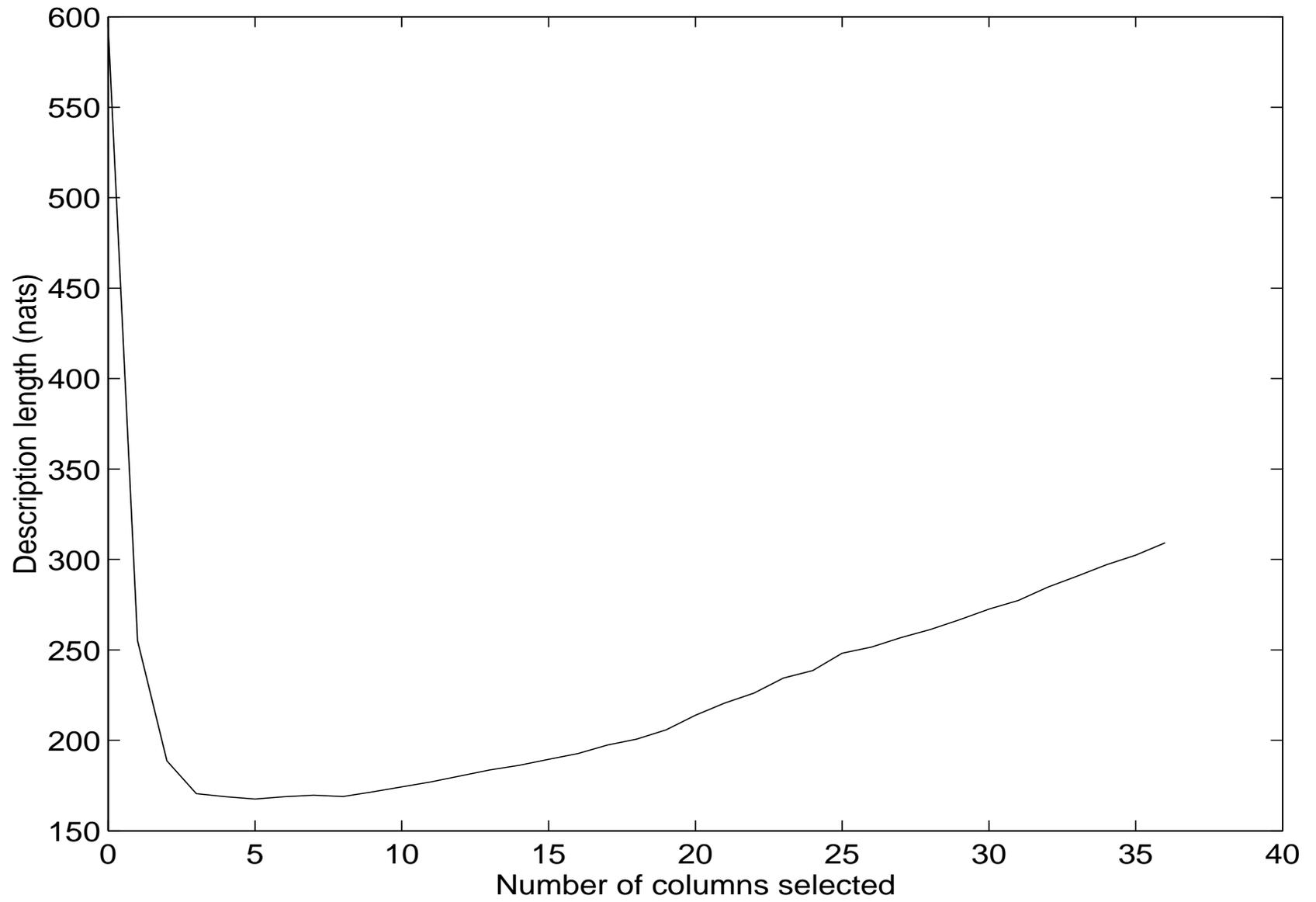
References

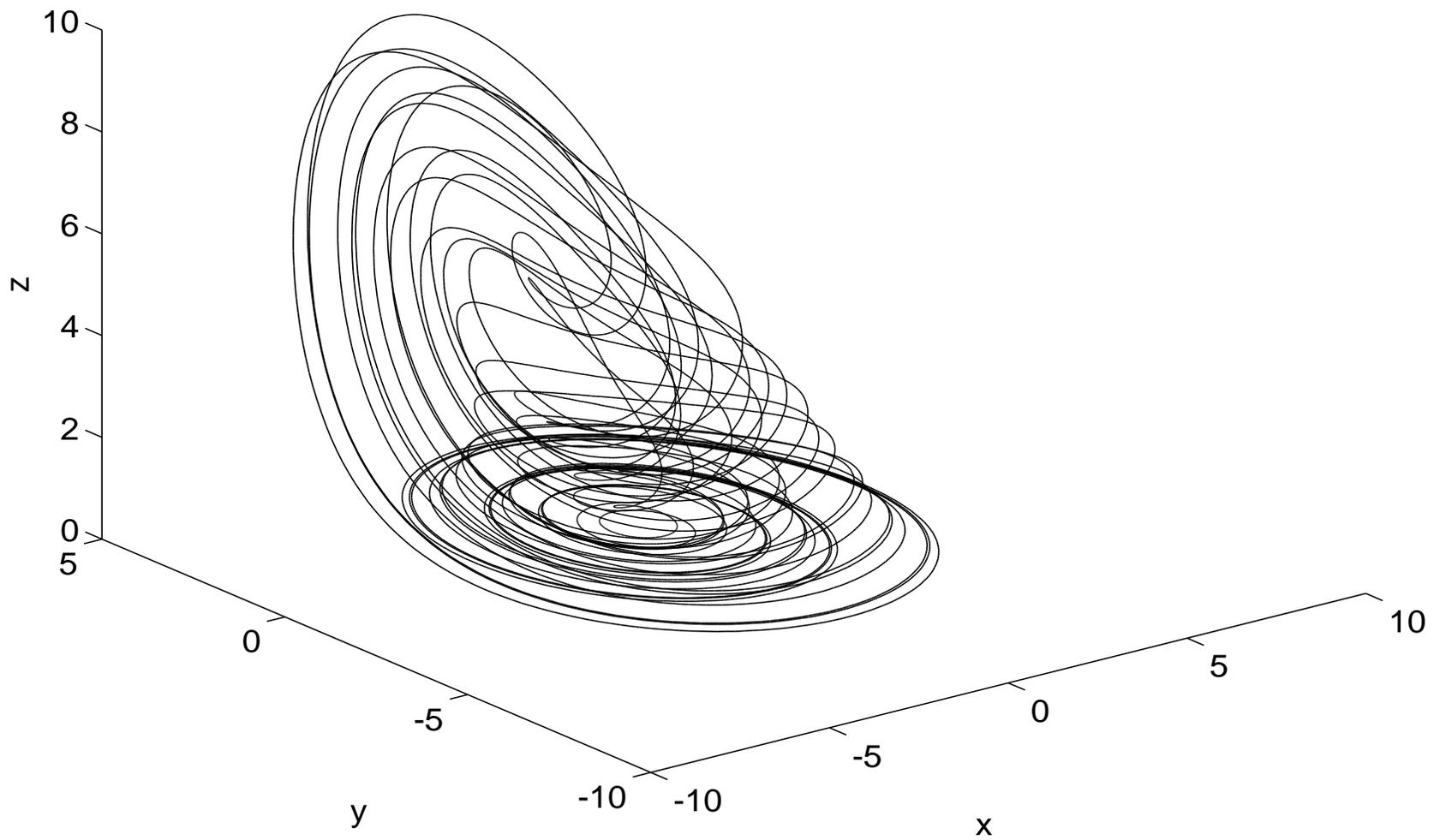
- [1] H. D. I. Abarbanel and M. B. Kennel. Local false nearest neighbors and dynamical dimensions from observed chaotic data. Technical report, Department of Physics, University of California, San Diego, 1992.
- [2] A. M. Albano, J. Muench, C. Schwartz, A. I. Mees, and P. E. Rapp. Singular value decomposition and the Grassberger-Procaccia algorithm. *Phys. Rev. A*, 38A:3017–3026, 1988.
- [3] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39:930–945, 1993.
- [4] S. Chen, C. F. N. Cowan, and P. M. Grant. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, 2:302–309, 1991.
- [5] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, San Francisco, 1979.
- [6] D. K. Ghaddar and H. Tong. Data transformation and self-exciting threshold autoregression. *J. R. Stat. Soc.*, C30:238–248, 1981.
- [7] P. Glendinning and C. T. Sparrow. Local and global behavior near homoclinic orbits. *J. Stat. Phys.*, 35:645–697, 1983.

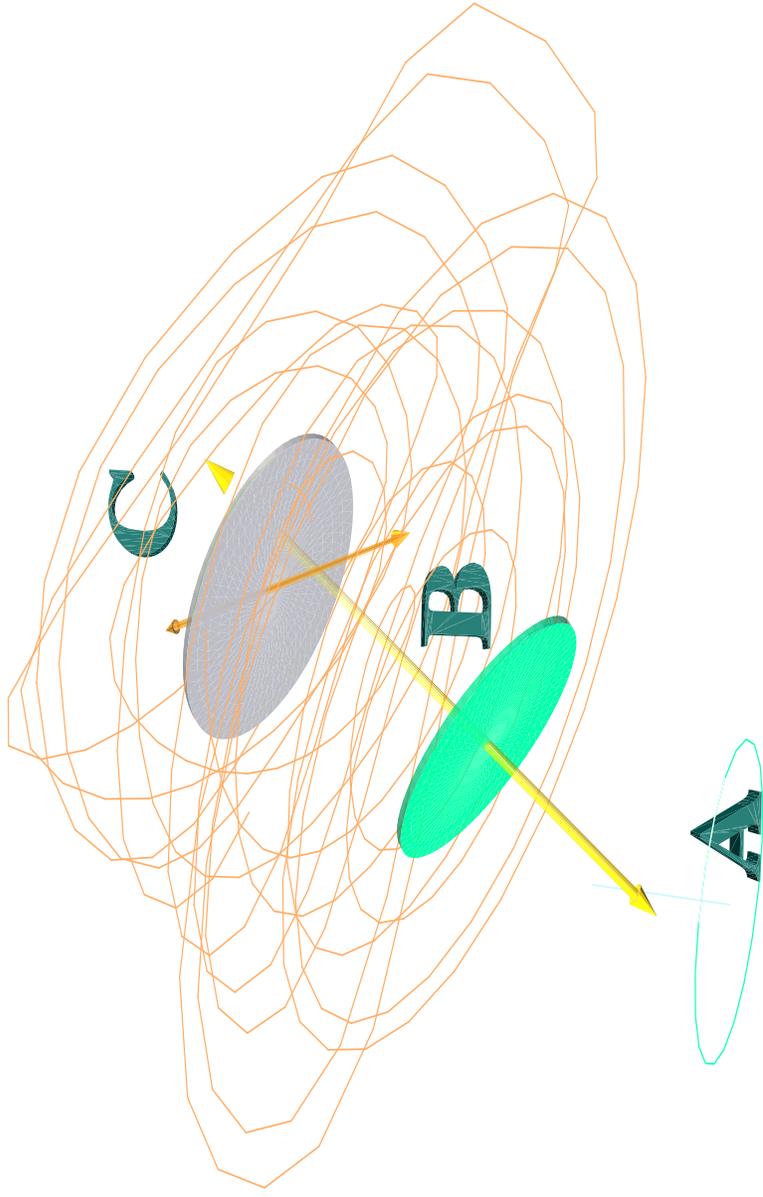
- [8] J. Glover and A. I. Mees. Reconstructing the dynamics of chua's circuit. *Journal of Circuits, Systems and Computers*, 3:201–214, 1992.
- [9] C. M. Goldie and R. G. E. Pinch. *Communication Theory*, volume 20. Cambridge University Press, Cambridge, 1991.
- [10] M. Kline. *Mathematical Thought from Ancient to Modern Times*. Oxford University Press, New York, 1972.
- [11] B. LeBaron. Persistence of the Dow Jones index on rising volume. Technical report, Santa Fe Institute, 1991.
- [12] J. A. Leonard and M. A. Kramer. Radial basis function networks for classifying process faults. *IEEE Control Systems*, April 1991:281–294, 1991.
- [13] A. I. Mees. Dynamical systems and tessellations: Detecting determinism in data. *International Journal of Bifurcation and Chaos*, 1:777–794, 1991.
- [14] A. I. Mees. Parsimonious dynamical reconstruction. *International Journal of Bifurcation and Chaos*, 3:669–675, 1993.
- [15] A. I. Mees. Reconstructing chaotic systems in the presence of noise. In *Proceedings of the 7th Toyota Conference, 1993: Towards the Harnessing of Chaos*. Elsevier, 1994.
- [16] A. I. Mees and R. K. Smith. Estimation and reconstruction in noisy chaotic systems. *In preparation*, 1993.
- [17] A. I. Mees and C. T. Sparrow. Some tools for analyzing chaos. *Proceedings IEEE*, 75:1058–1070, 1987.
- [18] K. Murty. A fundamental problem in linear inequalities with an application to tsp. *Math. Prog.*, 2:296–308, 1972.
- [19] L. Noakes. The Takens embedding theorem. *International Journal of Bifurcation and Chaos*, 1:867–872, 1991.
- [20] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*, volume 15. World Scientific, Singapore, 1989.
- [21] T. Sauer, J. A. Yorke, and M. Casdagli. Embedology. *Communications in Mathematical Physics*, Under consideration, 1991.
- [22] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.
- [23] S. Smale. Structurally stable systems are not dense. *Am. J. Math.*, 88, pp 491-496., 1966.
- [24] F. Takens. Detecting strange attractors in turbulence. In D. A. Rand and L. S. Young, editors, *Dynamical Systems and Turbulence*, volume 898, pages 365–381. Springer, Berlin, 1981.
- [25] H. Tong. *Non-linear Time Series: a Dynamical Systems Approach*. Oxford University Press, Oxford, 1990.
- [26] P. Whittle. *Optimization under constraints*. Wiley, Chichester, 1971.

Figure Captions

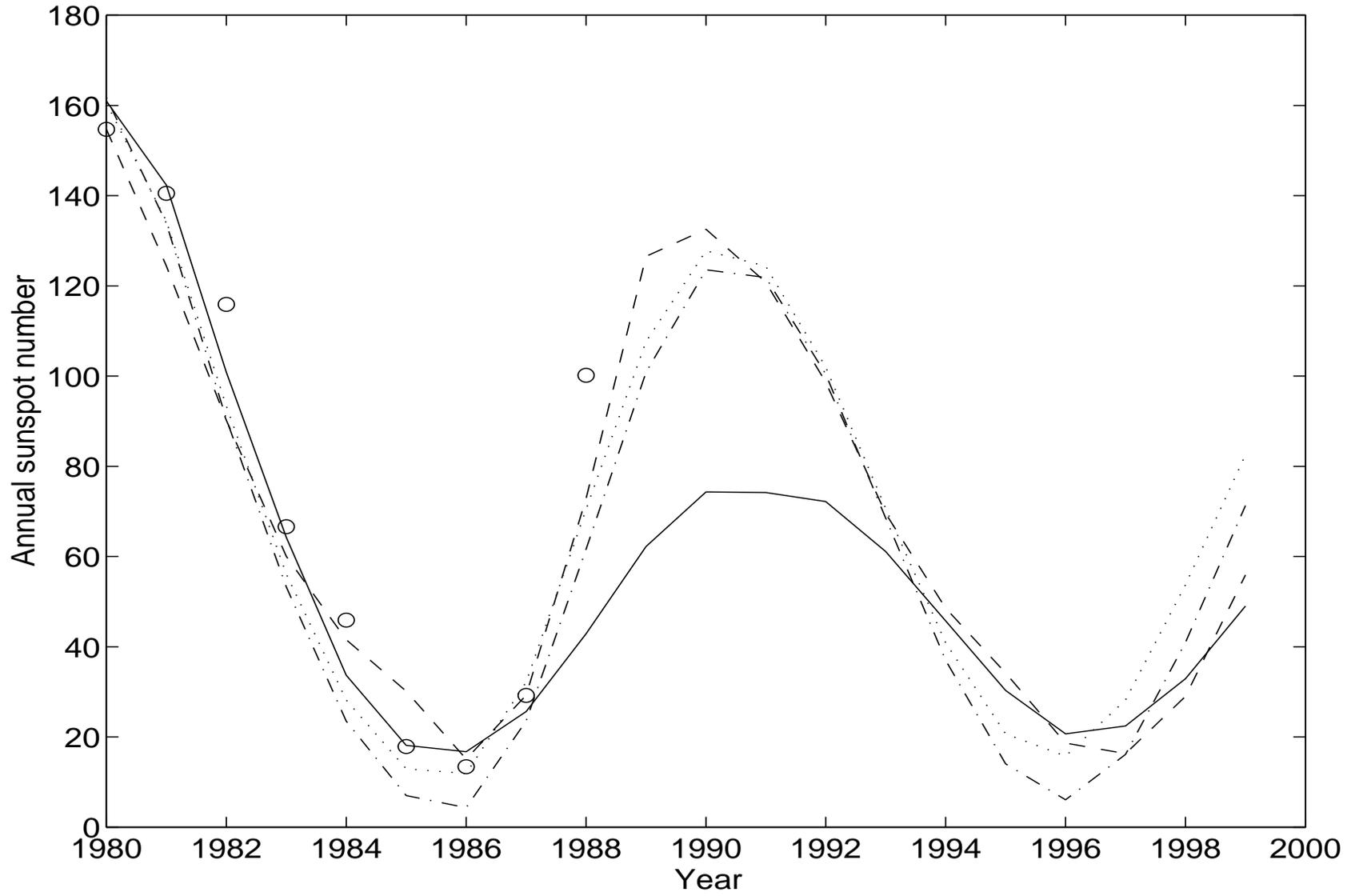
- Figure 1 A plot of description length against size of model using the selection algorithm described in Section 3. The data is the annual average sunspot number timeseries embedded in 3 dimensions with a lag of 1. The model was built from an initial set of constant and linear functions and 1000 radial basis functions, which were Gaussians with radius twice the standard deviation of the time-series and centers being chaperons of the data generated by adding 30% noise. The optimal model used the current year's sunspot number and 4 radial basis functions resulting in a model with an RMS error of 0.376 and a description length of 173 nats.
- Figure 2 An apparently chaotic motion of (18) when the parameters are $\sigma = 1$, $\omega = 15$, $\mu = 0.01$ and $\lambda = 6$.
- Figure 3 Embedded Shil'nikov x -coordinate data with fixed point locations estimated from a radial basis model. The discs and arrows are the invariant subspaces corresponding to the complex and real eigenvalues respectively. Fixed point B is implicated as the center of the Shil'nikov mechanism, while C is positioned where the flow escaping from B is folded back onto B's stable manifold. Fixed point A is the other fixed point of the system which has no role in the dynamics of this attractor, but nonetheless the model reconstructed from the timeseries predicts its existence.
- Figure 4 Predicted and actual annual sunspot numbers. Circles: actual annual sunspot numbers; solid line: piecewise linear SETAR model from Tong [25]; dash-dotted line: auto-regressive model of order 9 from Tong; dashed line: radial basis model by our algorithm; and dotted line: reduced auto-regressive model obtained by applying our algorithm to a purely linear set of basis functions.
- Figure 5 The annual average sunspot number time series and an artificial time series generated by an optimal radial basis model. The artificial time series is a section of a long time series generated from a random process (1), where \hat{F} is substituted for F and ϵ_t are Gaussian random variates with three-quarters the variance of the residuals of the fitted model. The artificial time series has many features, large and small, that are similar to features of the sunspot time series.



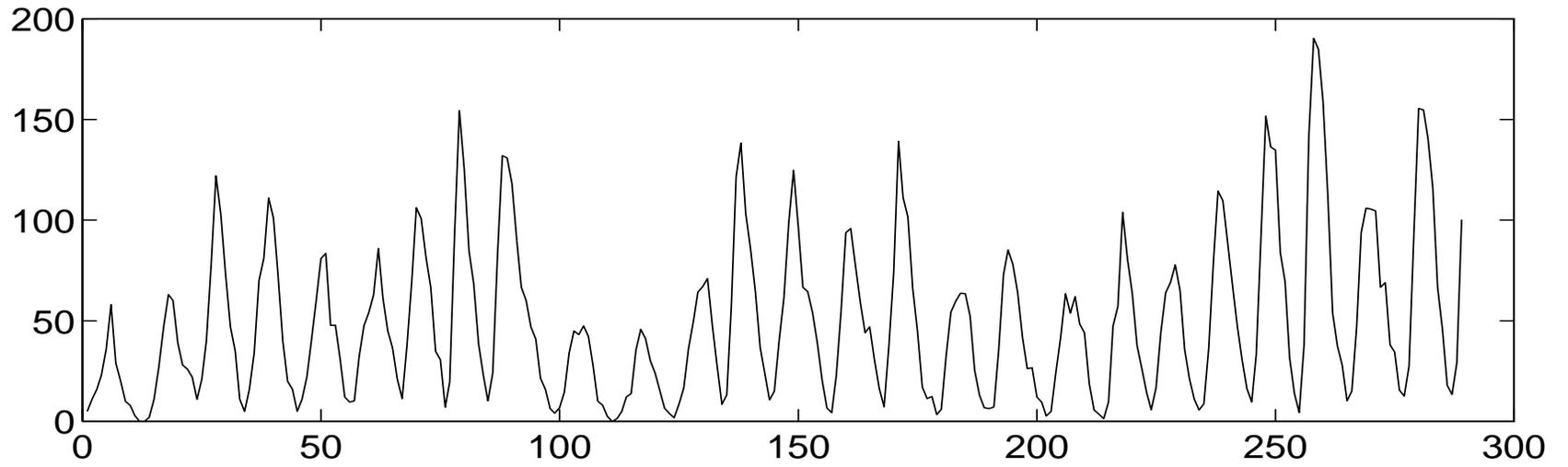




Predicted annual sunspot numbers



True sunspots series



Artificial time series

