

Chapter 3

Achieving Good Nonlinear Models: Keep It Simple, Vary the Embedding, and Get the Dynamics Right

Kevin Judd¹
Michael Small
Alistair I. Mees

ABSTRACT This chapter presents an overview of three fundamental notions in modeling nonlinear dynamical systems from time series. They are the use of the minimum description length (MDL) principle in model selection; the use of variable embedding and cylindrical basis models to build models that better capture the dynamics; and the use of $\Psi\Phi$ -models to eliminate systematic error when making long-term prediction. Their purposes are to separate what can be modeled (“determinism”) from what cannot (“noise”); to capture varying time-scales and different geometric features in embedding space; and to make models that have good long-term dynamical behavior as well as short-term predictive ability.

3.1 Introduction

If William of Ockham were alive today and were asked what to keep in mind when constructing a mathematical model of a dynamical system, he might answer, “Keep it simple. . .”. The danger of having an overly complex model is that it might display dynamical behavior totally unlike the behavior displayed by the system. One needs to adjust the complexity of a model so it can display most of the observed behavior of the system, while minimizing the potential of the model misbehaving. Our suggested method for doing this is to apply the principle of minimum description length (MDL), which is derived from information theory. We have found that MDL models capture dynamics better than models built without taking account of

¹Author for correspondence.

information theoretic aspects.

Constructing or fitting a model also implies working within some restricted *model class*. So another related issue is what class of models has sufficient complexity to capture the dynamical behavior of a large variety of systems. What we suggest is unusual. For some time it has been standard practice to first embed a time series (a Takens embedding), then to build a model (say, a radial basis model) in the embedded space. We introduce the notion of a *variable embedding*, which can be thought of as an embedding that changes with the state of the system. The equivalent of a radial basis model in a variable embedding scheme is a *cylindrical basis model*. The authors have found that variable embedding and cylindrical basis models capture dynamics better than standard uniform embedding and radial basis models. In short, cylinders stack up better than spheres.

A main focus in this chapter is the capturing of the dynamics of the system in a model, since this is a better measure of success than just prediction error. However, despite one's best efforts, a model will not be perfect and may still make systematic errors. The most likely cause of systematic errors (assuming everything else is done well) is that the "true" system does not lie in the model class one has chosen; that is, there is no model in the model class that has exactly the dynamics of the system. This most clearly shows itself when making long-term predictions. Even here there is something to be done; we suggest a simple technique of stacking models (called $\Psi\Phi$ -models [6]) that correct systematic errors and allow better long-term prediction with little additional effort.

3.2 Minimum Description Length Models: Keep It Simple

The minimum description length principle is an application of Ockham's Razor in a modeling context. It defines the best model for a time series to be the one that achieves the most concise description of the data. To understand how the principle works, suppose you (the "sender") have collected an experimental time series $x(t)$, $t = 1, \dots, n$ measured to an accuracy of (say) 12 bits and you wish to communicate this data to a colleague (the "recipient"). You could send the raw data. Alternatively, you could construct a dynamical model from the data that enables the recipient to predict a value of $x(t)$ from earlier values. If you and your colleague have previously agreed on a class of models, then you could communicate the data by sending the parameters of a model, enough initial data to start predicting future values of the time series, and the errors between the true time series and the values predicted by the model. Given this information, the recipient can reconstruct the experimental data to its full measured accuracy. An important point is that the parameters and errors need only

be specified to finite accuracy. Furthermore, if the model is good, then the total number of bits required to transmit parameters, initial values and errors will be less than the number of bits of raw data.

In practice the minimum description length principle requires calculating an approximation to the *description length* of the time series and model, which is effectively the number of bits required to transmit the model plus the number of bits required to transmit the errors. (The initial conditions are included in the parameter count, although their effect only matters when we are comparing different embedding dimensions.) Under fairly general assumptions one can write:

$$\begin{aligned} (\text{Description length}) \approx & \\ & (\text{number of data}) \times \log (\text{mean square prediction error}) \\ & + (\text{penalty for number and accuracy of parameters}). \end{aligned}$$

As the number of parameters in a model increases the (in-sample) prediction errors decrease, but eventually, the penalty for introducing another parameter outweighs the benefit it has in reducing (the in-sample) prediction errors. The model that attains the minimum description length is the optimal model within the class of models considered. We do not have space here to discuss in detail why this is successful; extensive discussions are to be found elsewhere [10, 4, 14].

In special model classes, explicit approximations to the description length can be calculated. A particularly useful class of parameterized nonlinear autoregressive model consists of those we call *pseudo-linear* models, also called general linear models, which have the form

$$x(t+1) = \sum_{i=1}^m \lambda_i f_i(v(t)) + \epsilon_t, \quad (3.1)$$

$$v(t) = (x(t), x(t-1), \dots, x(t-d)) \quad (3.2)$$

for some selection of nonlinear functions f_i , unknown parameters λ_i and unknown independent and identically distributed random variates ϵ_t . (Observe in passing that choosing $v(t)$ amounts to using a particular embedding.) Define

$$V_i = (f_i(v(1)), \dots, f_i(v(n)))^T, i = 1, \dots, m, \quad (3.3)$$

$$y = (x(1), \dots, x(n))^T, \quad (3.4)$$

$$\lambda = (\lambda_1, \dots, \lambda_m)^T, \quad (3.5)$$

and let V be the matrix whose columns are V_i , $i = 1, \dots, m$. If the ϵ_t are assumed to be Gaussian and λ has been chosen to minimize the sum of squares of the prediction errors $e = y - V\lambda$, then [4] the description length

is bounded by

$$\left(\frac{n}{2} - 1\right) \ln \frac{e^T e}{n} + (k + 1) \left(\frac{1}{2} + \ln \gamma\right) - \sum_{j=1}^k \ln \delta_j, \quad (3.6)$$

where k is the number of non-zero components of λ , γ is related to the scale of the data (for example, a constant used to scale the observations to lie in the unit interval) and δ solves $[Q\delta]_j = 1/\delta_j$ where

$$Q = \hat{V}^T \hat{V} / e^T e$$

and \hat{V} is composed of just those columns of V that correspond to nonzero elements of λ . The variables δ can be interpreted as the relative precision to which the parameters λ are specified.

The attraction of pseudo-linear models is that the parameters λ are easily calculated, because the sum of squares of the prediction errors $e = y - V\lambda$ can be minimized efficiently using singular value decomposition or any of its many equivalents. What makes general pseudo-linear models different from, and more powerful than, special cases such as linear or global polynomial models, is that the basis functions f_i can be chosen in many ways.

The critical problem is how to select the basis functions f_i . In general these will be nonlinear functions depending on various additional parameters that should be optimized over. Unfortunately, this optimization is nonlinear and so is in general difficult, slow and prone to capture by local minima. (This problem is well known in modeling via single-layer neural nets, a particular pseudo-linear approach.) Instead of optimizing the parameters of a few basis functions, we can generate many fixed basis functions, not only at the start but also adaptively as the model building progresses, and select a subset of them that optimizes the description length.

This alternative scheme requires an efficient combinatorial optimization method to select an optimal subset of the basis functions. It would appear that we have made the problem worse, because combinatorial optimization is notoriously hard, but in fact the following subset selection algorithm, described in detail elsewhere [4] is very successful in most of the applications we have considered. The algorithm selects subsets that are near-optimal according to the description length criterion and hence produces good pseudo-linear models. It operates by adding and removing candidate functions from a given basis set according to a local optimality criterion, and accepting a set of given size as optimal if the same candidate is removed as was just added. The size of the basis set is increased until the description length criterion says it has become too large, and then the best set found so far is selected as the overall optimum.

In the algorithm, B represents any set of $k < m$ indices in $\{1, \dots, m\}$. We write V_B for the $n \times k$ matrix formed from the columns of V with indices in B , λ_B for the least squares solution to $y = V_B \lambda$, and $e_B = y - V_B \lambda_B$.

Algorithm 1

1. Normalize the columns of V to have unit length.
2. Let $S_0 = (\frac{n}{2} - 1) \ln(y^T y/n) + \frac{1}{2} + \ln \gamma$ (the description length of the raw data).
3. Let $B = \{j\}$ where V_j is the column of V such that $|V_j^T y|$ is maximum (this selects as the first basis function the one that most closely matches the data y ; note that $\lambda_B = V_j^T y/V_j^T V_j$ in this case).
4. Let $\mu = V^T e_B$ and i be the index of the component of μ with maximum absolute value. Let $B' = B \cup \{i\}$ (the components of the vector μ measure how closely each of the basis functions not currently in use will match the error of the current model; extend the current model with the basis function that best matches the current error).
5. Calculate $\lambda_{B'}$. Let o be the index in B' corresponding to the component of $\lambda_{B'}$ with smallest absolute value (here o is the index of the basis function that makes the smallest contribution to the current extended model).
6. If $i \neq o$, then put $B = B' \setminus \{o\}$ and go to step 4. Otherwise, set $B = B'$. (Throw out the “worst” basis function o if it is not i , the last one we brought in; then go back and try again. Otherwise, the extended basis B' is taken to be the “locally” optimal basis.)
7. Define $B_k = B$, where $k = |B|$. Find δ such that $(V_B^T V_B \delta)_j = 1/\delta_j$ for each $j = \{1, \dots, k\}$ and calculate $S_k = (\frac{n}{2} - 1) \ln \frac{\hat{\epsilon}^T \hat{\epsilon}}{n} + (k + 1)(\frac{1}{2} + \ln \gamma) - \sum_{j=1}^k \ln \hat{\delta}_j$. (At this stage we have found the best model of size k that can be built from the best model of size $k - 1$ by “bringing in the best and throwing out the worst.”)
8. If $S_k < S_{k-1}$, then go to step 4. (Continue until the description length stops decreasing.)
9. Take the basis B_k such that S_k is minimum as the optimal model.

3.3 Variable Embedding: Cylinders Stack Up Better than Spheres

For some time it has been a standard practice to embed a time series (a Takens embedding by delay reconstruction), then build a model (say, a radial basis model) in the embedded space. Unfortunately, there are many types of dynamical behavior that are not modeled well by this technique.

3.3.1 Uniform Embedding

Given a time series $x(t) \in \mathbb{R}$ one might form an embedded time series

$$z(t) = (x(t), x(t - \ell), \dots, x(t - (d - 1)\ell)) \in \mathbb{R}^d$$

where ℓ is called the *lag*. In anticipation of our discussion we will refer to this embedding as a *uniform* embedding. The lag is introduced to improve the observability, for example, of noisy time series. The lag is chosen to optimize the spread of the embedded time series without confusing the dynamics and to obtain an embedding that is independent of the sampling rate (of an oversampled continuous time system, for example). There are two principal methods of choosing the lag: the first zero of the autocorrelation function [1] and the minimum of the mutual information [3].

Uniform embeddings for modeling purposes are at their most effective when embedding a time series with a single dominant periodicity or recurrence time. Both of the above mentioned methods for calculating lags give similar lags for such time series and the lag is approximately one-quarter of the dominant period. For this lag the embedded time series is ring-shaped; shorter and longer lags result in elliptical rings, or—if the lags are too far from a good value—a scrambled mess. A good lag in this case keeps states that correspond to similar phases close together and anti-phase states as far apart as possible. Uniform embeddings are quite suitable for classic chaotic systems such the Rössler and Lorenz systems, which have a single dominant periodicity or recurrence time.

3.3.2 Nonuniform Embedding

Uniform embeddings can fail when there are multiple strong periodicities with differing time scales. For example, consider a quasi-periodic time series with differing frequency components or with very close frequencies that lead to a “carrier” frequency and a “modulation” of differing period. Figure 3.1 shows three time series, from different systems, that all possess short and long period recurrences. Uniform embedding fails for such time series because a short lag would be optimal for the high-frequency component and a long lag would be optimal for the low-frequency components and modulation, while a compromise lag is inadequate for both time scales.

One way to avoid this problem is to use nonuniform embedding strategies. For example, with the sunspot time series one would most likely choose to use the uniform embedding $(x(t), x(t - 3), x(t - 6))$, but the authors find that the embedding $(x(t), x(t - 2), x(t - 8))$ gives better results when constructing radial basis models [4, 5]. Note how compared to the uniform embedding the nonuniform embedding has two components that are more closely spaced and yet a broader window overall; alternatively, one could say the uniform embedding is a compromise between a shorter lag and a wider window.

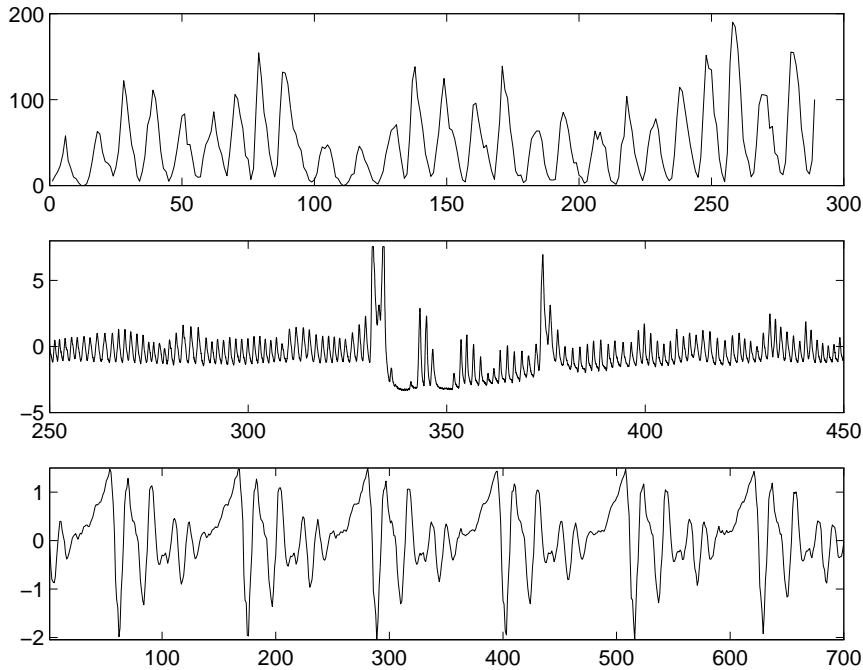


FIGURE 3.1. Three time series that have multiple, strong periodicities: (a) the average annual sunspot number; (b) a sampling at 50Hz of a measurement that is proportional to the cross-sectional area of the abdomen of a child during quiet sleep: the phenomenon observed here is called “periodic breathing”; (c) a 12kHz recording of the Japanese vowel [a].

One can use a nonuniform embedding in any situation where one uses a uniform embedding; there is just a little additional bookkeeping required to ensure enough of the past time series is retained and that correct observations are used to make each forward prediction. Of course, to make long-term predictions or simulations of the dynamics, the prediction step must be a divisor of *all* the lags.

3.3.3 Variable Embedding

Going a step further, we suggest it is often advantageous to vary the embedding strategy with the state of the system. To visualize why it is useful to do this, consider modeling the Lorenz system with its butterfly-shaped attractor. When the system state is out on the “wings” of the attractor, a two-dimensional embedding is sufficient to model and predict motions—one would require a lot of very high-quality data to discern the thickness

of these wings. However, near the origin where the crossover of the wings occurs, a three-dimensional embedding is essential. One could imagine constructing a perfectly adequate model that does not use a global embedding but rather uses appropriate local embeddings as the system state varies.

If one uses a variable embedding, then the processes of embedding and modeling are merged into one process with a single optimization goal of finding a compact and accurate model. An example of this, and more generally how variable embedding can be implemented, is the class of *cylindrical basis models*.

3.3.4 Cylindrical Basis Models

The standard radial basis model [8, 2, 7] is pseudo-linear, with each of the nonlinear functions depending only on the radial distance from a certain point called a center. That is, in the standard pseudo-linear model of equation 3.1, the functions have the form $f_i(z) = \phi(|z - c_i|/r_i)$ for suitably chosen centers c_i , radii r_i and function ϕ . If the function ϕ is decreasing, we can think of the action of each f_i localized on a ball. If we ignore some coordinates (as a result of a local nonuniform embedding) then the functions act locally on cylinders, instead of radially symmetric spheres, so a reasonable name is *cylindrical basis models*. To construct such models we must define the various cylinders.

A cylinder is defined by a center c_i , a radius r_i and a lag vector $(\ell_1, \ell_2, \dots, \ell_k)$. The lag vector corresponds to a projection P such that

$$P(v(t)) = (x(t - \ell_1), x(t - \ell_2), \dots, x(t - \ell_k)).$$

The basis functions

$$f_i(z) = \phi(|P_i(z - c_i)|/r_i), \quad (3.7)$$

with decreasing ϕ , have the effect of localizing the embedding in a cylindrical neighborhood of c_i , where the axis of the cylinder is parallel to the components of $v(t)$ that have been projected out, that is, the lags that are missing in $P(v(t))$. To simplify the notation we have introduced a redundancy in (3.7), because the center c_i is unique only up to the projection P_i . A very simple example of a cylindrical basis function given $(x, y, z) \in \mathbb{R}^3$ a Gaussian radial basis function with center $(2, -1, 5)$ and radius 3 would be

$$f(x, y, z) = e^{((x-2)^2 + (y+1)^2 + (z-5)^2)/18}.$$

Whereas given a projection $P(x, y, z) = (x, z)$ there is a cylindrical basis function

$$f(x, y, z) = e^{((x-2)^2 + (z-5)^2)/18}.$$

To construct a cylindrical basis model using the selection methods we have described, one would generate a large set of basis functions (3.7), with

decreasing function ϕ , having different centers c_i , radii r_i and projections P_i (which define the lag vectors). There is an obvious combinatorial explosion here because for each potential center there are $2^d - 1$ possible lag vectors. This could be tackled using genetic algorithms or simulated annealing, but we have found that a relatively simple adaptation of our earlier algorithms appears to avoid the worst of this explosion.

Algorithm 2

1. Let S represent an initial set of candidate basis functions. These functions are likely to be generated randomly but possibly using some additional information to choose likely candidates for selection, for example, a selection weighted by the errors of the current best model. One good way to start is to generate centers and radii as usual and then apply the reduced autoregressive method *locally* in the region around each center to get a local lag vector. This lag vector is only an initial guess; it will typically be thinned later in the process.
2. Apply Algorithm 1 to determine the best model using the basis functions of S . Let S^* be the selected basis functions.
3. If desired, locally optimize S^* by tuning its parameters using some standard method such as the Levenberg-Marquardt algorithm [9].
4. Generate a new set of candidate basis functions S that includes S^* . The new candidate functions might be chosen with additional knowledge gained from the selection of S^* . (For example, we might try simplifying cylinders by applying addition projections, randomly perturbing selected cylinders, or putting new cylinders near where the model makes its worst predictions.)
5. Return to step 2, and continue to do so while S^* is changing (significantly).

The authors have found that cylindrical basis models are much more successful at capturing the dynamics of a system than radial basis models [5, 11, 12].

3.4 Systematic Errors: $\Psi\Phi$ -Models

One of the important reasons for wanting to model the dynamics correctly is for making long-term predictions. One can obtain longer-term predictions from a short-term predictor by simply iterating the predictor. Such longer-term predictions are limited in their accuracy by observational and dynamic noise and the sensitivity to initial conditions of the dynamical system, but iterated predictors are also limited by systematic errors in the short-term

predictor, which can arise from under- or overfitting data, or from the model class not containing the system under study.

There are many reasons why a model can have systematic errors. The model may have been derived from physical or chemical principles but various assumptions may not hold in practice, or a dynamic model may lack sufficient resolution, or the exact nature of the noise may be unknown. For example, even an analogue circuit or mechanical device [13] designed to instantiate the Lorenz equations would not be modeled exactly by the Lorenz equations, because the diodes and capacitors, or mechanical devices, will have slightly different nonlinearities. Indeed it is likely that there is no easily described transcendental functions that model the apparatus exactly. This implies that there would not be any reasonable model class that exactly describes the system; the system is outside all reasonable model classes and any reasonable model will display some systematic error.

In principle, with black box and other models, one could make a model increasingly more complex to account for the systematic errors, but there comes a point when it is more productive to abandon this adaptive approach that merely adds more components of similar type and try something entirely different in character. What we suggest here is not to throw out a reasonably good model, but to apply a method that augments and corrects it cheaply and effectively [6]. The just cited paper discusses in more detail how the method we are about to introduce has the effect of greatly expanding the general linear model class in a way that is useful for modeling dynamical systems. Another way of looking at the method is that it tries to rearrange and present the information in a time series in a more useful or accessible way.

It is an important observation that, from an information theory perspective, iterated predictors have a fundamental flaw. Given a time series x_1, \dots, x_t , we can think of a one-step predictor φ as attempting to add an additional datum \hat{x}_{t+1} to the end of the time series, where

$$\hat{x}_{t+1} = \varphi(x_t, x_{t-1}, \dots, x_{t-d+1}) = x_{t+1} + \text{error}. \quad (3.8)$$

Iterated prediction to obtain further \hat{x}_{t+m} requires shifting the arguments of φ one place to the right (removing x_{t-d+1}) and substituting the predicted \hat{x}_{t+1} into the first position, and so on. Observe that each subsequent iterated prediction uses no more information than was used in the first one-step prediction. From an information theory point of view there is a mistake here, the longer-term predictions ought to use *more* information. The $\Psi\Phi$ -method proposed in this chapter attempts to correct this failure by exploiting information retained in the systemic errors of iterated predictions.

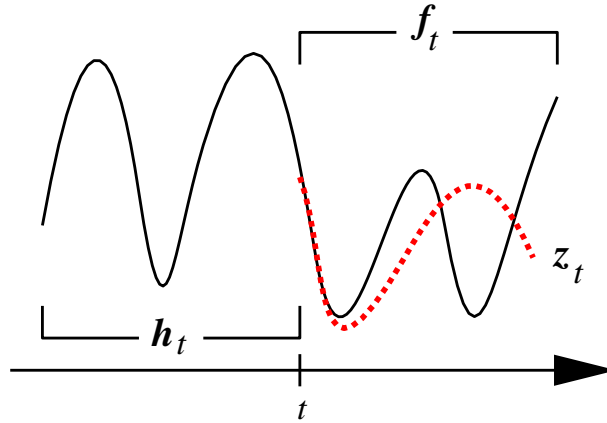


FIGURE 3.2. Schematic of a time series (solid line) and the sections of the time series that comprise the *history* vector h_t and the *future* vector f_t . The dotted line represents an iterated predictions z_t with systematic error.

Referring to Figure 3.2, define

$$\begin{aligned} h_t &= (x_t, x_{t-1}, \dots, x_{t-d+1}) \in \mathbb{R}^d, \\ f_t &= (x_{t+1}, \dots, x_{t+p}) \in \mathbb{R}^p, \\ z_t &= (\hat{x}_{t+1}, \dots, \hat{x}_{t+q}) \in \mathbb{R}^q, \end{aligned}$$

where \hat{x}_{t+m} is the m th iterated prediction of φ , using the predict, shift, substitute method. The vector h_t represents the important recent *history* of the time series at time t , and f_t is the next p steps of *future* after time t . The vector z_t is our predictions of the next q steps obtained by iterated prediction with φ .²

Ideally we want a mapping $h_t \mapsto f_t$, but we only have a map

$$z_t = \Phi(h_t), \quad (3.9)$$

which gives an estimate z_t of f_t ; if $p = q$, then

$$f_t = z_t + \text{errors}.$$

If there is observational or dynamic noise, or if there is overfitting or underfitting, or if the system is not in our model class, then the predictions z_t of the future f_t may be inaccurate or even erroneous. Also the predictions are incomplete when $q < p$.

²How we obtained the predictions is not important: they could be obtained from a physical model, a black-box model, or a multi-step predictor, or they could even be predictions from several different models.

To correct the errors and inaccuracies of the *predictor* Φ we propose finding a *corrector* Ψ ,

$$f_t = \Psi(z_t) + \text{smaller errors.} \quad (3.10)$$

The essential point here is that if the errors of Φ are *systematic*, then Ψ can exploit this information to produce a better prediction of f_t . We shall see that Φ can be quite a poor predictor, but provided it is consistent, a Ψ mapping can improve it greatly.

We will refer to the method just described as the $\Psi\Phi$ -method³. The surprising thing is that this method works well, even for simple Ψ , yielding consistently good results for both artificial and experimental systems using cubic polynomial maps for Ψ . Radial basis models have also been used to obtain marginally better results than we show here, but on occasions even a linear or quadratic Ψ can work well.

Figure 3.3 shows long-term predictions obtained for the Lorenz system with $\approx 15\%$ observational noise using the $\Psi\Phi$ -method; a detailed discussion of this figure and other experimental results can be found elsewhere [6].

$\Psi(z_t)$ can best be thought of as a prediction of the mean long-term behavior of the system. This can be seen in Figure 3.3 where it can sometimes happen that so much critical information about the system has been lost that it is no longer possible to predict which “wing” of the Lorenz attractor the system will be on, and consequently $\Psi(z_t)$ predicts that the system’s mean behavior is zero. This is, of course, the correct prediction, as disappointing as it may seem. On the other hand, in Figure 3.3 it can be seen that the estimated prediction error (upper and lower solid lines) of this conservative estimate may vary in a way that demonstrates that not all dynamic information has been lost.

Determining the Corrector Ψ

We now describe how to determine a suitable corrector Ψ and how to estimate prediction errors of the $\Psi\Phi$ -map. We will only consider the case where Ψ is linear in its parameters, for example, linear, polynomial, radial basis, pseudo-linear and general linear, because such mappings are easy to work with and appear to be adequate in all cases we have examined.

The length p of the future vector f_t may be greater than the length q of the iterated prediction vector z_t , so that Ψ extrapolates beyond the end of z_t ; indeed Ψ could also interpolate by using an alternative definition

$$z_t = (\hat{x}_{t+s_1}, \hat{x}_{t+s_2}, \dots, \hat{x}_{t+s_k}) \in \mathbb{R}^k,$$

where $0 < s_1 < \dots < s_k \leq q$. This “thinned out” alternative for z_t is

³ $\Psi\Phi$ is pronounced *sci fi*, in the “To boldly go...” sense.

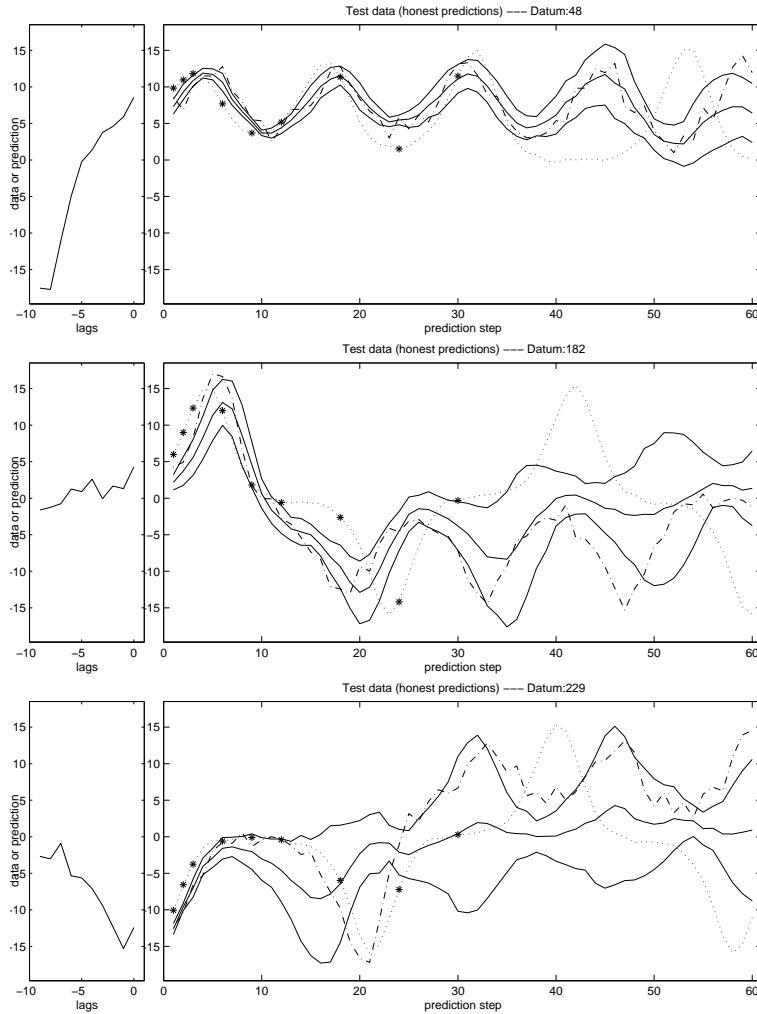


FIGURE 3.3. Three indicative long-term predictions obtained for the Lorenz system with $\approx 15\%$ observational noise using the $\Psi\Phi$ -method. The model was built from 4000 points sampled at 0.05 second. The test data was a distinct trajectory. The left panels show h_t ; the right panels show the future f_t (dash-dotted line); the iterated Φ -model predictions (dotted line) with components of $\Phi(h_t)$ used in z_t indicated by stars, i.e. a thinned prediction vector was used; the $\Psi\Phi$ -model predictions (central solid line) and the Θ predictions of absolute deviations (upper and lower solid lines).

useful to reduce the effort involved in computing Ψ . Assume, without loss of generality that $k \leq q \leq p$, which is the natural thing to do.

Given a time series $x = \{x_t : -d + 1 \leq t \leq n + p\}$ one can construct vec-

tors h_t and f_t for $t = 1, \dots, n$. Then for each h_t one can calculate a z_t that is an *in-sample* long-term prediction of the f_t . Any discrepancies between z_t and the corresponding components of f_t are clues to the systematic errors of Φ at h_t ; see Figure 3.2.

Given n vectors z_t and f_t the construction and estimation of the mapping Ψ is easily achieved when Ψ is linear in its parameters. If, for example, Ψ is also linear in the components of z_t , then Ψ can be defined by a $p \times q$ matrix A (or $p \times k$ for the alternative definition of z_t) such that,

$$\Psi(z_t) = Az_t. \quad (3.11)$$

In general, if Ψ is a linear combination of m *basis* functions $g_i(z)$, $i = 1, \dots, m$, then

$$\Psi(z_t) = A\zeta_t \quad (3.12)$$

where $\zeta_t = (g_1, \dots, g_m)(z_t) \in \mathbb{R}^m$ and A is some $p \times m$ matrix. For example, a polynomial Ψ has as basis functions the constant function, projections onto each component of z_t , and the products of powers of the components of z_t up to the order of the polynomial.

Estimation of the parameter matrix A is straightforward using least squares. If one chooses to minimize the sum of squares of prediction errors $\|f_t - \Psi(z_t)\|$, then this corresponds to solving the following matrix equation in the least squares sense (by singular value decomposition, for example)

$$F = AZ, \quad (3.13)$$

where F is the $p \times n$ matrix having the future vector f_t as the t th column, and Z is the $m \times n$ matrix whose t th column is ζ_t . One could conceivably want to use a weighted norm when calculating $\|f_t - \Psi(z_t)\|$, since the standard Euclidean norm weights all prediction errors equally, regardless of how far into the future; however, we have used the simple length norm.

The procedure is summarized in the following algorithm.

Algorithm 3

1. From a given time series, form the vector time series h_t and f_t .
2. For each “initial condition” h_t iterate the model Φ (whatever that model might be) to form the predictions vector z_t .
3. Extend the vector z_t to include powers and products of the components (optionally, use some other set of basis function).
4. Pack the column vectors f_t and z_t into matrices F and Z .
5. Solve $F = AZ$.
6. To make a long-term prediction, iterate the Φ model from an initial condition and multiply this vector of predictions into A .

Estimating Prediction Errors

It is desirable to predict how good the $\Psi\Phi$ -predictions are, because this essentially estimates the prediction horizon in advance. Because we do not anticipate a necessarily Gaussian distribution of prediction errors, and because it is easier to calculate, we will attempt to predict the *robust* estimator of spread given by the mean absolute differences, rather than attempting to predict the variance of the prediction errors.

Define the absolute prediction error vector e_t by

$$e_t = |f_t - \Psi(z_t)| \in \mathbb{R}^p \quad (3.14)$$

where $|\cdot|$ applies componentwise.

We now propose to estimate a mapping $\Theta: \mathbb{R}^q \rightarrow \mathbb{R}^p$ (or from \mathbb{R}^k for the alternative definition of z_t) such that $\Theta(z_t)$ estimates e_t . There is no reason to estimate this map any differently from how we proposed estimating Ψ , that is, choose Θ to be linear in its parameters and so

$$\Theta(z_t) = B\zeta_t, \quad (3.15)$$

where ζ_t is the basis functions evaluations as before, although possibly a different set of basis functions, and B is some $p \times m$ matrix. Estimating Θ by least squares as before requires solving the matrix equation in the least squares sense

$$E = BZ \quad (3.16)$$

where the t th column of E is e_t the error given in (3.14).

3.5 Conclusions

The focus of this chapter has been the development of reliable methods of modeling nonlinear dynamical systems from time series, so that the model captures the dynamics. This involves using the minimum description length principle to ensure that models neither underfit nor overfit data, using variable embeddings to locally optimize models to the dynamics, and using $\Psi\Phi$ -models to correct any residual systematic errors when making long-term predictions.

Acknowledgments

The work of AIM was partially supported by a grant from the Australian Research Council. AIM also thanks the Department of Systems Engineering at The Chinese University of Hong Kong for hospitality.

References

- [1] A. M. Albano, A. I. Mees, G. C. deGuzman, and P. E. Rapp. Data requirements for reliable estimation of correlation dimensions. In H. Degn, A. V. Holden, and L. F. Olsen, editors, *Chaos in biological systems*, pages 207–220. Plenum, New York, 1987.
- [2] M. Casdagli. Nonlinear prediction of chaotic time series. *Physica D*, 35(3):335–356, 1989.
- [3] A. M. Fraser and H. L. Swinney. Independent coordinates for strange attractors from mutual information. *Physical Review A*, 33(2):1134–1140, 1986.
- [4] K. Judd and A. I. Mees. On selecting models for nonlinear time series. *Physica D*, 82:426–444, 1995.
- [5] K. Judd and A. I. Mees. Embedding as a modeling problem. *Physica D*, 120:273–286, 1998.
- [6] K. Judd and M. Small. Towards long-term prediction. *Physica D*, 136:31–44, 2000.
- [7] A. I. Mees. Parsimonious dynamical reconstruction. *International Journal of Bifurcation and Chaos*, 3(3):669–675, 1993.
- [8] A. I. Mees, M. F. Jackson, and L. O. Chua. Device modeling by radial basis functions. *IEEE Trans CAS/FTA*, 39(1):19–27, 1992.
- [9] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, Cambridge, 1988.
- [10] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*, volume 15 of *Series in Computer Science*. World Scientific, Singapore, 1989.
- [11] M. Small and K. Judd. Detecting nonlinearity in experimental data. *International Journal of Bifurcation and Chaos*, 8(6):1231–1244, 1997.
- [12] M. Small and K. Judd. Comparisons of new nonlinear modeling techniques with applications to infant respiration. *Physica D*, 117:283–298, 1998.
- [13] S. H. Strogatz. *Nonlinear Dynamics and Chaos*. Addison Wesley, New York, 1994.
- [14] P. Vitanyi and M. Li. Ideal MDL and its relation to Bayesianism. In D. L. Dowe, K. B. Korb, and J. J. Oliver, editors, *Information, Statistics and Induction in Science*, pages 282–291, Melbourne, Australia, 1996. World Scientific.