The Future of Time Series

Neil A. Gershenfeld

MIT Media Lab
20 Ames Street
Cambridge, MA 02139

Andreas S. Weigend

Computer Science Dept
University of Colorado
Boulder, CO 80309–0430

Citation:
N. A. Gershenfeld and A. S. Weigend, "The Future of Time Series." In: *Time Series Prediction: Forecasting the Future and Understanding the Past*, A. S. Weigend and N. A. Gershenfeld, eds., 1–70. Addison-Wesley, 1993.

Preface

This book is the result of an unsuccessful joke. During the summer of 1990, we were both participating in the Complex Systems Summer School of the Santa Fe Institute. Like many such programs dealing with "complexity," this one was full of exciting examples of how it can be possible to recognize when apparently complex behavior has a simple understandable origin. However, as is often the case in young disciplines, little effort was spent trying to understand how such techniques are interrelated, how they relate to traditional practices, and what the bounds on their reliability are. These issues must be addressed if suggestive results are to grow into a mature discipline. Problems were particularly apparent in time series analysis, an area that we both arrived at in our respective physics theses. Out of frustration with the fragmented and anecdotal literature, we made what we thought was a humorous suggestion: run a competition. Much to our surprise, no one laughed and, to our further surprise, the Santa Fe Institute promptly agreed to support it. The rest is history (630 pages worth).

Reasons why a competition might be a bad idea abound: science is a thoughtful activity, not a simple race; the relevant disciplines are too dissimilar and the questions too difficult to permit meaningful comparisons; and the required effort might be prohibitively large in return for potentially misleading results. On the other hand, regardless of the very different techniques and language games of the different disciplines that study time series (physics, biology, economics,...), very

similar questions are asked: What will happen next? What kind of system produced the time series? How can it be described? How much can we know about the system? These questions can have quantitative answers that permit direct comparisons. And with the growing penetration of computer networks, it has become feasible to announce a competition, to distribute the data (withholding the continuations), and subsequently to collect and analyze the results. We began to realize that a competition might not be such a crazy idea.

The Santa Fe Institute seemed ideally placed to support such an undertaking. It spans many disciplines and addresses broad questions that do not easily fall within the purview of a single academic department. Following its initial commitment, we assembled a group of advisors[1] to represent many of the relevant disciplines in order to help us decide if and how to proceed. These initial discussions progressed to the collection of a large library of candidate data sets, the selection of a representative small subset, the specification of the competition tasks, and finally the publicizing and then running of the competition (which was remotely managed by Andreas in Bangkok and Neil in Cambridge, Massachusetts). After its close, we ran a NATO Advanced Research Workshop to bring together the advisory board, representatives of the groups that had provided the data, successful participants, and interested observers. This heterogeneous group was able to communicate using the common reference of the competition data sets; the result is this book. It aims to provide a snapshot of the range of new techniques that are currently used to study time series, both as a reference for experts and as a guide for novices.

Scanning the contents, we are struck by the variety of routes that lead people to study time series. This subject, which has a rather dry reputation from a distance (we certainly thought that), lies at the heart of the scientific enterprise of building models from observations. One of our goals was to help clarify how new time series techniques can be broadly applicable beyond the restricted domains within which they evolved (such as simple chaos experiments), and, at the same time, how theories of everything can be applicable to nothing given the limitations of real data.

We had another hidden agenda in running this competition. Any one such study can never be definitive, but our hope was that the real result would be planting a seed for an ongoing process of using new technology to share results in what is, in effect, a very large collective research project. The many papers in this volume that use the competition tasks as starting points for the broader and deeper study of these common data sets suggests that our hope might be fulfilled. This survey of what is possible is in no way meant to suggest that better results are impossible. We will be pleased if the Santa Fe data sets and results become common reference benchmarks, and even more pleased if they are later discarded and replaced by more worthy successors.

[1] The advisors were Leon Glass (biology), Clive Granger (economics), Bill Press (astrophysics and numerical analysis), Maurice Priestley (statistics), Itamar Procaccia (dynamical systems), T. Subba Rao (statistics), and Harry Swinney (experimental physics).

An undertaking such as this requires the assistance of more friends (and thoughtful critics) than we knew we had. We thank the members of the advisory board, the providers of the data sets, and the competition entrants for participating in a quixotic undertaking based on limited advance information. We thank the Santa Fe Institute and NATO for support.[2] We are grateful for the freedom provided by Stanford, Harvard, Chulalongkorn, MIT, and Xerox PARC. We thank Ronda Butler-Villa and Della Ulibarri for the heroic job of helping us assemble this book, and we thank the one hundred referees for their critical comments. We also thank our friends for not abandoning us despite the demands of this enterprise.

Finally, we must thank each other for tolerating and successfully filtering each other's occasionally odd ideas about how to run a time series competition, which neither of us would have been able to do (or understand) alone.

Neil Gershenfeld
Cambridge, MA

Andreas Weigend
San Francisco, CA

July 1993

[2] Core funding for the Santa Fe Institute is provided by the John D. and Catherine T. MacArthur Foundation, the National Science Foundation, grant PHY-8714918, and the U.S. Department of Energy, grant ER-FG05-88ER25054.

TABLE OF CONTENTS

	Abstract	1
1	Introduction	2
2	The Competition	4
3	Linear Time Series Models	11
	3.1 ARMA, FIR, and all that	11
	3.2 The Breakdown of Linear Models	16
4	Understanding and Learning	18
	4.1 Understanding: State-Space Reconstruction	20
	4.2 Learning: Neural Networks	25
5	Forecasting	30
	5.1 State-Space Forecasting	30
	5.2 Connectionist Forecasting	34
	5.3 Beyond Point-Predictions	36
6	Characterization	42
	6.1 Simple Tests	42
	6.2 Direct Characterization via State Space	45
	6.3 Indirect Characterization: Understanding through Learning	53
7	The Future	60
	Appendix	63
	Appendix to the Book: Accessing the Server	
	References	

Neil A. Gershenfeld† and Andreas S. Weigend‡
†MIT Media Laboratory, 20 Ames Street, Cambridge, MA 02139;
e-mail: neilg@media.mit.edu.
‡Xerox PARC, 3333 Coyote Hill Road, Palo Alto, CA 94304;
e-mail: weigend@cs.colorado.edu.
Address after August 1993: Andreas Weigend, Department of Computer Science and Institute of Cognitive Science, University of Colorado, Boulder, CO 80309-0430.

The Future of Time Series: Learning and Understanding

Throughout scientific research, measured time series are the basis for characterizing an observed system and for predicting its future behavior. A number of new techniques (such as state-space reconstruction and neural networks) promise insights that traditional approaches to these very old problems cannot provide. In practice, however, the application of such new techniques has been hampered by the unreliability of their results and by the difficulty of relating their performance to those of mature algorithms. This chapter reports on a competition run through the Santa Fe Institute in which participants from a range of relevant disciplines applied a variety of time series analysis tools to a small group of common data sets in order to help make meaningful comparisons among their approaches. The design and the results of this competition are described, and the historical and theoretical backgrounds necessary to understand the successful entries are reviewed.

Times Series Prediction: Forecasting the Future and Understanding the Past,
A. S. Weigend and N. A. Gershenfeld, eds. Reading, MA: Addison-Wesley, 1993.

1. INTRODUCTION

The desire to predict the future and understand the past drives the search for laws that explain the behavior of observed phenomena; examples range from the irregularity in a heartbeat to the volatility of a currency exchange rate. If there are known underlying deterministic equations, in principle they can be solved to forecast the outcome of an experiment based on knowledge of the initial conditions. To make a forecast if the equations are not known, one must find both the rules governing system evolution and the actual state of the system. In this chapter we will focus on phenomena for which underlying equations are not given; the rules that govern the evolution must be inferred from regularities in the past. For example, the motion of a pendulum or the rhythm of the seasons carry within them the potential for predicting their future behavior from knowledge of their oscillations without requiring insight into the underlying mechanism. We will use the terms "understanding" and "learning" to refer to two complementary approaches taken to analyze an unfamiliar time series. *Understanding* is based on explicit mathematical insight into how systems behave, and *learning* is based on algorithms that can emulate the structure in a time series. In both cases, the goal is to explain observations; we will not consider the important related problem of using knowledge about a system for controlling it in order to produce some desired behavior.

Time series analysis has three goals: forecasting, modeling, and characterization. The aim of *forecasting* (also called *predicting*) is to accurately predict the short-term evolution of the system; the goal of *modeling* is to find a description that accurately captures features of the long-term behavior of the system. These are not necessarily identical: finding governing equations with proper long-term properties may not be the most reliable way to determine parameters for good short-term forecasts, and a model that is useful for short-term forecasts may have incorrect long-term properties. The third goal, system *characterization*, attempts with little or no *a priori* knowledge to determine fundamental properties, such as the number of degrees of freedom of a system or the amount of randomness. This overlaps with forecasting but can differ: the complexity of a model useful for forecasting may not be related to the actual complexity of the system.

Before the 1920s, forecasting was done by simply extrapolating the series through a global fit in the time domain. The beginning of "modern" time series prediction might be set at 1927 when Yule invented the autoregressive technique in order to predict the annual number of sunspots. His model predicted the next value as a weighted sum of previous observations of the series. In order to obtain "interesting" behavior from such a linear system, outside intervention in the form of external shocks must be assumed. For the half-century following Yule, the reigning paradigm remained that of linear models driven by noise.

However, there are simple cases for which this paradigm is inadequate. For example, a simple iterated map, such as the logistic equation (Eq. (11), in Section 3.2), can generate a broadband power spectrum that cannot be obtained by a linear approximation. The realization that apparently complicated time series can be generated by very simple equations pointed to the need for a more general theoretical framework for time series analysis and prediction.

Two crucial developments occurred around 1980; both were enabled by the general availability of powerful computers that permitted much longer time series to be recorded, more complex algorithms to be applied to them, and the data and the results of these algorithms to be interactively visualized. The first development, state-space reconstruction by time-delay embedding, drew on ideas from differential topology and dynamical systems to provide a technique for recognizing when a time series has been generated by deterministic governing equations and, if so, for understanding the geometrical structure underlying the observed behavior. The second development was the emergence of the field of machine learning, typified by neural networks, that can adaptively explore a large space of potential models. With the shift in artificial intelligence from rule-based methods towards data-driven methods,[1] the field was ready to apply itself to time series, and time series, now recorded with orders of magnitude more data points than were available previously, were ready to be analyzed with machine-learning techniques requiring relatively large data sets.

The realization of the promise of these two approaches has been hampered by the lack of a general framework for the evaluation of progress. Because time series problems arise in so many disciplines, and because it is much easier to describe an algorithm than to evaluate its accuracy and its relationship to mature techniques, the literature in these areas has become fragmented and somewhat anecdotal. The breadth (and the range in reliability) of relevant material makes it difficult for new research to build on the accumulated insight of past experience (researchers standing on each other's toes rather than shoulders).

Global computer networks now offer a mechanism for the disjoint communities to attack common problems through the widespread exchange of data and information. In order to foster this process and to help clarify the current state of time series analysis, we organized the Santa Fe Time Series Prediction and Analysis Competition under the auspices of the Santa Fe Institute during the fall of 1991. The goal was not to pick "winners" and "losers," but rather to provide a structure for researchers from the many relevant disciplines to compare quantitatively the results of their analyses of a group of data sets selected to span the range of studied problems. To explore the results of the competition, a NATO Advanced Research Workshop was held in the spring of 1992; workshop participants included members of the competition advisory board, representatives of the groups that had collected the data, participants in the competition, and interested observers. Although the participants came from a broad range of disciplines, the discussions were framed by

[1] Data sets of hundreds of megabytes are routinely analyzed with massively parallel supercomputers, using parallel algorithms to find near neighbors in multidimensional spaces (K. Thearling, personal communication, 1992; Bourgoin et al., 1993).

the analysis of common data sets and it was (usually) possible to find a meaningful common ground. In this overview chapter we describe the structure and the results of this competition and review the theoretical material required to understand the successful entries; much more detail is available in the articles by the participants in this volume.

2. THE COMPETITION

The planning for the competition emerged from informal discussions at the Complex Systems Summer School at the Santa Fe Institute in the summer of 1990; the first step was to assemble an advisory board to represent the interests of many of the relevant fields.[2] With the help of this group we gathered roughly 200 megabytes of experimental time series for possible use in the competition. This volume of data reflects the growth of techniques that use enormous data sets (where automatic collection and processing is essential) over traditional time series (such as quarterly economic indicators, where it is possible to develop an intimate relationship with each data point).

In order to be widely accessible, the data needed to be distributed by ftp over the Internet, by electronic mail, and by floppy disks for people without network access. The latter distribution channels limited the size of the competition data to a few megabytes; the final data sets were chosen to span as many of a desired group of attributes as possible given this size limitation (the attributes are shown in Figure 2). The final selection was:

A. **A clean physics laboratory experiment.** 1,000 points of the fluctuations in a far-infrared laser, approximately described by three coupled nonlinear ordinary differential equations (Hübner et al., this volume).

B. **Physiological data from a patient with sleep apnea.** 34,000 points of the heart rate, chest volume, blood oxygen concentration, and EEG state of a sleeping patient. These observables interact, but the underlying regulatory mechanism is not well understood (Rigney et al., this volume).

C. **High-frequency currency exchange rate data.** Ten segments of 3,000 points each of the exchange rate between the Swiss franc and the U.S. dollar. The average time between two quotes is between one and two minutes (Lequarré, this volume). If the market was efficient, such data should be a random walk.

[2]The advisors were Leon Glass (biology), Clive Granger (economics), Bill Press (astrophysics and numerical analysis), Maurice Priestley (statistics), Itamar Procaccia (dynamical systems), T. Subba Rao (statistics), and Harry Swinney (experimental physics).

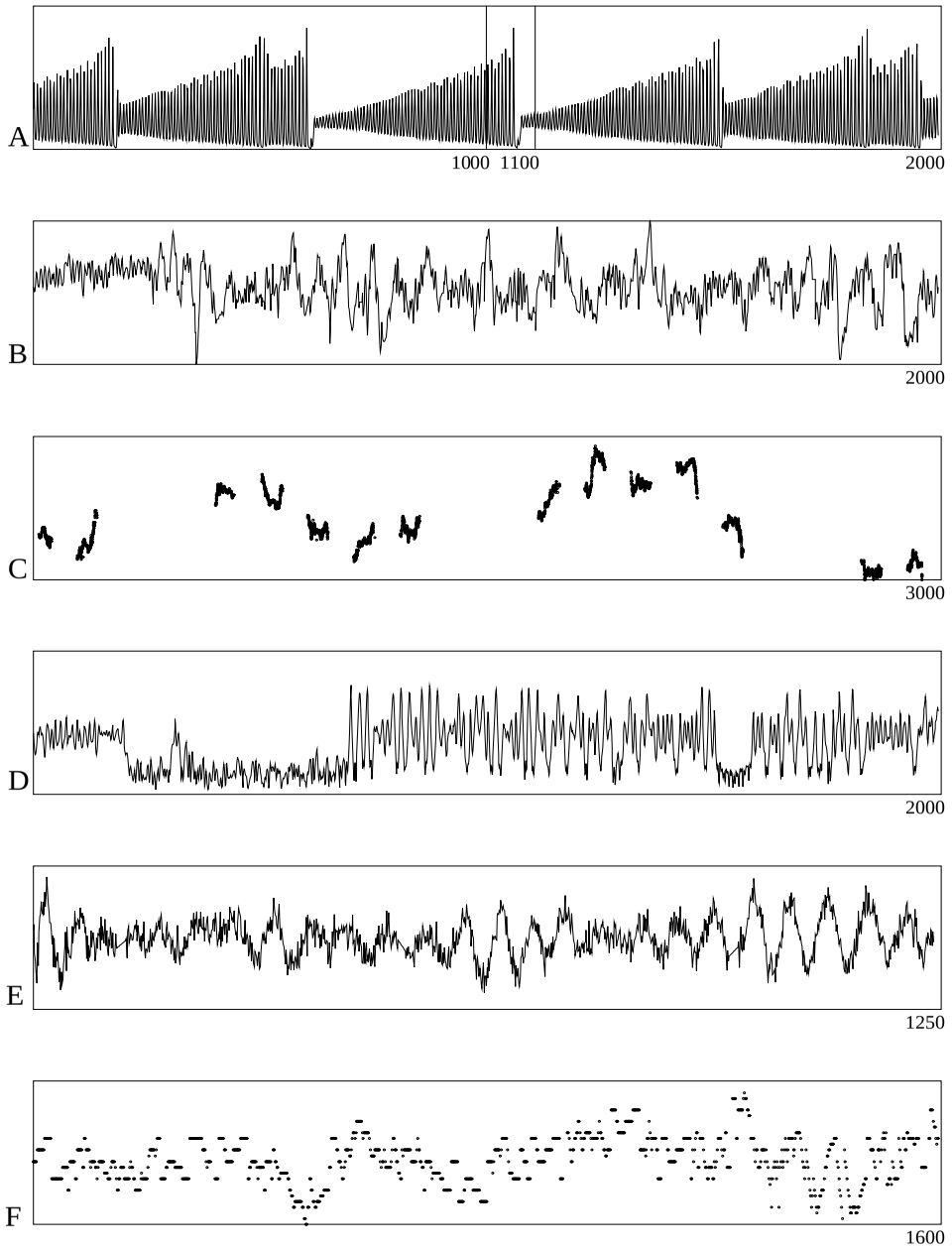

FIGURE 1 Sections of the competition data sets.

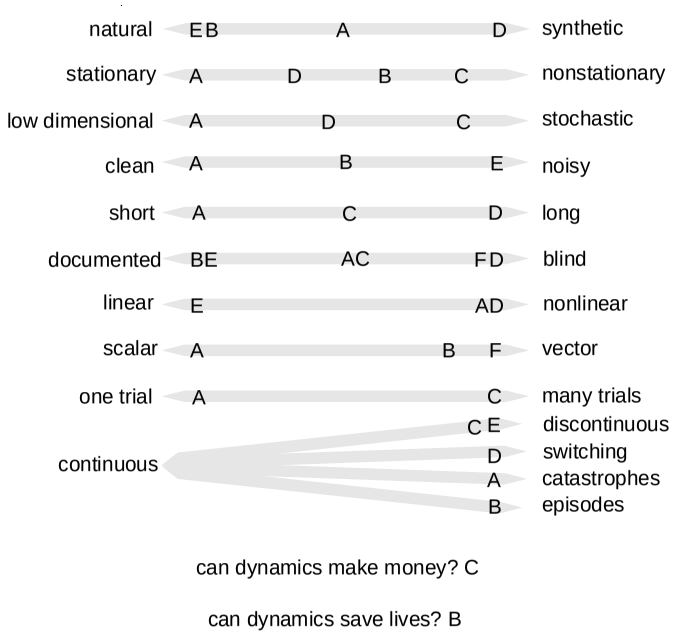

FIGURE 2 Some attributes spanned by the data sets.

D. **A numerically generated series designed for this competition.** A driven particle in a four-dimensional nonlinear multiple-well potential (nine degrees of freedom) with a small nonstationarity drift in the well depths. (Details are given in the Appendix.)

E. **Astrophysical data from a variable star.** 27,704 points in 17 segments of the time variation of the intensity of a variable white dwarf star, collected by the *Whole Earth Telescope* (Clemens, this volume). The intensity variation arises from a superposition of relatively independent spherical harmonic multiplets, and there is significant observational noise.

F. **A fugue.** J. S. Bach's final (unfinished) fugue from *The Art of the Fugue*, added after the close of the formal competition (Dirst and Weigend, this volume).

The amount of information available to the entrants about the origin of each data set varied from extensive (Data Sets B and E) to blind (Data Set D). The original files will remain available. The data sets are graphed in Figure 1, and some of the characteristics are summarized in Figure 2. The appropriate level of description for models of these data ranges from low-dimensional stationary dynamics to stochastic processes.

After selecting the data sets, we next chose **competition tasks** appropriate to the data sets and research interests. The participants were asked to:

- predict the (withheld) continuations of the data sets with respect to given error measures,
- characterize the systems (including aspects such as the number of degrees of freedom, predictability, noise characteristics, and the nonlinearity of the system),
- infer a model of the governing equations, and
- describe the algorithms employed.

The data sets and competition tasks were made publicly available on August 1, 1991, and competition entries were accepted until January 15, 1992. Participants were required to describe their algorithms. (Insight in some previous competitions was hampered by the acceptance of proprietary techniques.) One interesting trend in the entries was the focus on prediction, for which three motivations were given: (i) because predictions are falsifiable, insight into a model used for prediction is verifiable; (ii) there are a variety of financial incentives to study prediction; and (iii) the growth of interest in machine learning brings with it the hope that there can be universally and easily applicable algorithms that can be used to generate forecasts. Another trend was the general failure of simplistic "black-box" approaches—in all successful entries, exploratory data analysis preceded the algorithm application.[3]

It is interesting to compare this time series competition to the previous state of the art as reflected in two earlier competitions (Makridakis & Hibon, 1979; Makridakis et al., 1984). In these, a very large number of time series was provided (111 and 1001, respectively), taken from business (forecasting sales), economics (predicting recovery from the recession), finance, and the social sciences. However, all of the series used were very short, generally less than 100 values long. Most of the algorithms entered were fully automated, and most of the discussion centered around linear models.[4] In the Santa Fe Competition all of the successful entries were fundamentally nonlinear and, even though significantly more computer power was used to analyze the larger data sets with more complex models, the application of the algorithms required more careful manual control than in the past.

[3] The data, analysis programs, and summaries of the results are available by anonymous ftp from `ftp.santafe.edu`, as described in the Appendix to this volume. In the competition period, on average 5 to 10 people retrieved the data per day, and 30 groups submitted final entries by the deadline. Entries came from the U.S., Europe (including former communist countries), and Asia, ranging from junior graduate students to senior researchers.

[4] These discussions focused on issues such as the order of the linear model. Chatfield (1988) summarizes previous competitions.

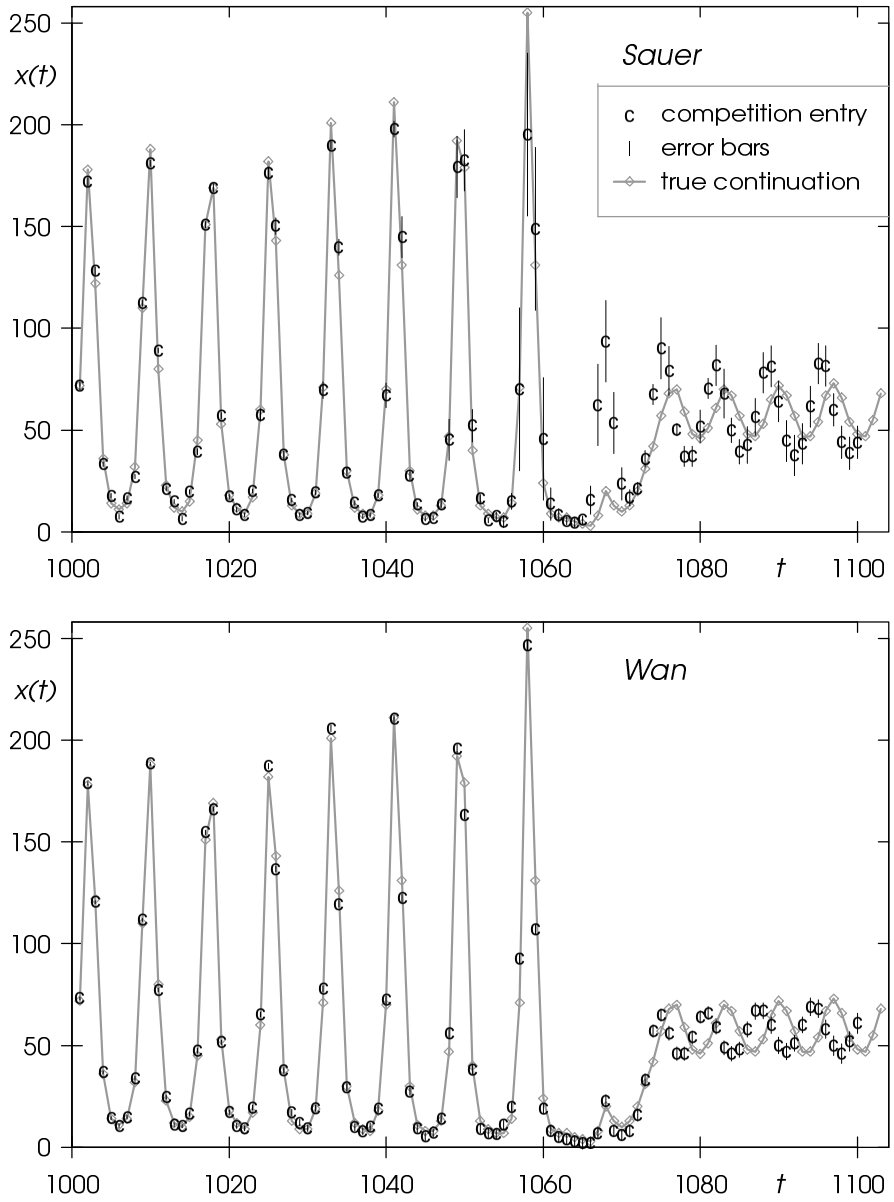

FIGURE 3 The two best predicted continuations for Data Set A, by Sauer and by Wan. Predicted values are indicated by "c," predicted error bars by vertical lines. The true continuation (not available at the time when the predictions were received) is shown in grey (the points are connected to guide the eye).

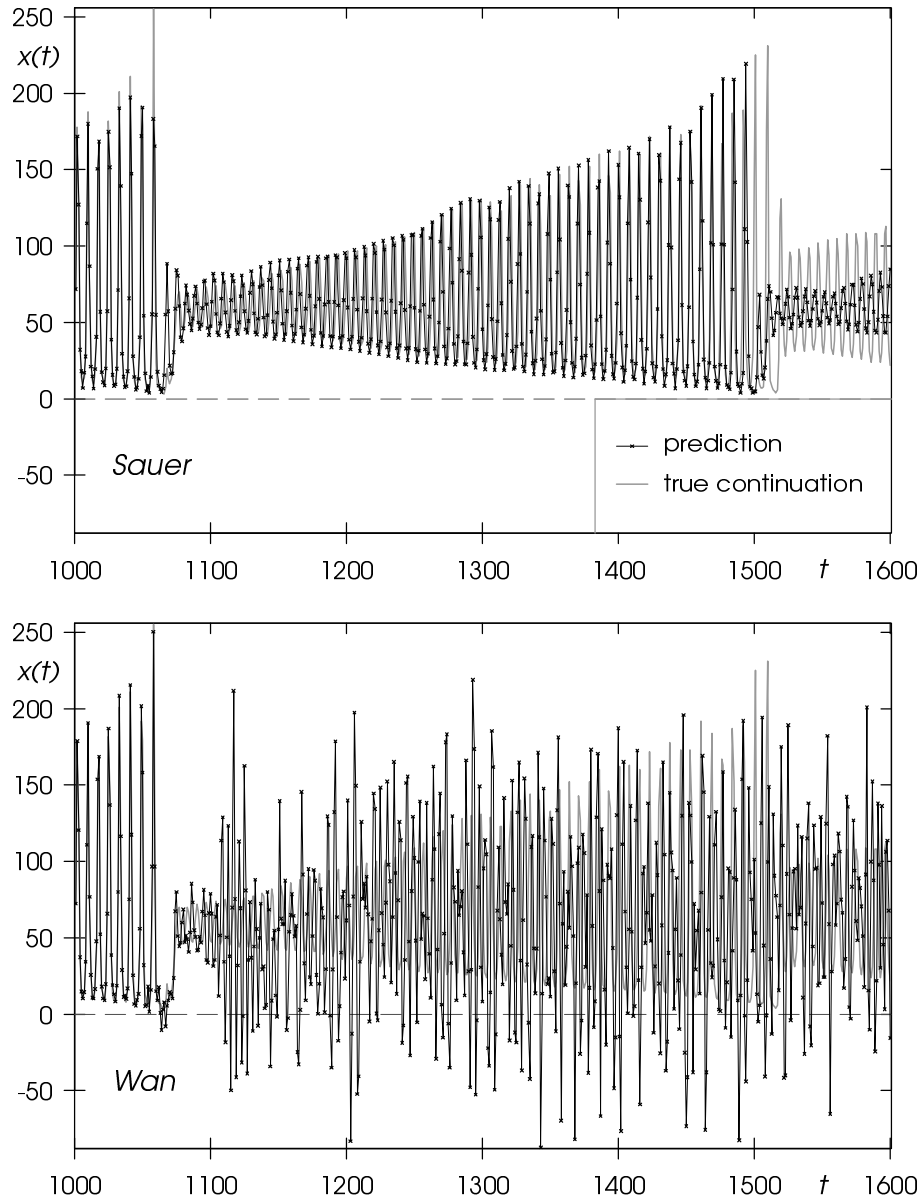

FIGURE 4 Predictions obtained by the same two models as in the previous figure, but continued 500 points further into the future. The solid line connects the predicted points; the grey line indicates the true continuation.

As an example of the results, consider the intensity of the laser (Data Set A; see Figure 1). On the one hand, the laser can be described by a relatively simple "correct" model of three nonlinear differential equations, the same equations that Lorenz (1963) used to approximate weather phenomena. On the other hand, since the 1,000-point training set showed only three of four collapses, it is difficult to predict the next collapse based on so few instances.

For this data set we asked for predictions of the next 100 points as well as estimates of the error bars associated with these predictions. We used two measures to evaluate the submissions. The first measure (normalized mean squared error) was based on the predicted values only; the second measure used the submitted error predictions to compute the likelihood of the observed data given the predictions. The Appendix to this chapter gives the definitions and explanations of the error measures as well as a table of all entries received. We would like to point out a few interesting features. Although this single trial does not permit fine distinctions to be made between techniques with comparable performance, two techniques clearly did much better than the others for Data Set A; one used state-space reconstruction to build an explicit model for the dynamics and the other used a connectionist network (also called a neural network). Incidentally, a prediction based solely on visually examining and extrapolating the training data did much worse than the best techniques, but also much better than the worst.

Figure 3 shows the two best predictions. Sauer (this volume) attempts to understand and develop a *representation* for the geometry in the system's state space, which is the best that can be done without knowing something about the system's governing equations, while Wan (this volume) addresses the issue of *function approximation* by using a connectionist network to learn to emulate the input-output behavior. Both methods generated remarkably accurate predictions for the specified task. In terms of the measures defined for the competition, Wan's squared errors are one-third as as large as Sauer's, and—taking the predicted uncertainty into account—Wan's model is four times more likely than Sauer's.[5] According to the competition scores for Data Set A, this puts Wan's network in the first place.

A different picture, which cautions the hurried researcher against declaring one method to be universally superior to another, emerges when one examines the evolution of these two prediction methods further into the future. Figure 4 shows the same two predictors, but now the continuations extend 500 points beyond the 100 points submitted for the competition entry (no error estimates are shown).[6] The neural network's class of potential behavior is much broader than what can be generated from a small set of coupled ordinary differential equations, but the state-space model is able to reliably forecast the data much further because its explicit description can correctly capture the character of the long-term dynamics.

[5] The likelihood ratio can be obtained from Table 2 in the Appendix as $\exp(-3.5)/\exp(-4.8)$.

[6] Furthermore, we invite the reader to compare Figure 5 by Sauer (this volume, p. 191) with Figure 13 by Wan (this volume, p. 213). Both entrants start the competition model at the same four (new) different points. The squared errors are compared in the Table on p.192 of this book.

In order to understand the details of these approaches, we will detour to review the framework for (and then the failure of) linear time series analysis.

3. LINEAR TIME SERIES MODELS

Linear time series models have two particularly desirable features: they can be understood in great detail and they are straightforward to implement. The penalty for this convenience is that they may be entirely inappropriate for even moderately complicated systems. In this section we will review their basic features and then consider why and how such models fail. The literature on linear time series analysis is vast; a good introduction is the very readable book by Chatfield (1989), many derivations can be found (and understood) in the comprehensive text by Priestley (1981), and a classic reference is Box and Jenkins' book (1976). Historically, the general theory of linear predictors can be traced back to Kolmogorov (1941) and to Wiener (1949).

Two crucial assumptions will be made in this section: the system is assumed to be linear and stationary. In the rest of this chapter we will say a great deal about relaxing the assumption of linearity; much less is known about models that have coefficients that vary with time. To be precise, unless explicitly stated (such as for Data Set D), we assume that the underlying equations do not change in time, i.e., *time invariance* of the system.

3.1 ARMA, FIR, AND ALL THAT

There are two complementary tasks that need to be discussed: understanding how a given model behaves and finding a particular model that is appropriate for a given time series. We start with the former task. It is simplest to discuss separately the role of external inputs (moving average models) and internal memory (autoregressive models).

3.1.1 PROPERTIES OF A GIVEN LINEAR MODEL.

Moving average (MA) models. Assume we are given an external input series $\{e_t\}$ and want to modify it to produce another series $\{x_t\}$. Assuming linearity of the system and causality (the present value of x is influenced by the present and N past values of the input series e), the relationship between the input and output is

$$x_t = \sum_{n=0}^{N} b_n e_{t-n} = b_0 e_t + b_1 e_{t-1} + \cdots + b_N e_{t-N} \,. \qquad (1)$$

This equation describes a convolution filter: the new series x is generated by an Nth-order filter with coefficients $b_0, \cdots, b_n$ from the series e. Statisticians and econometricians call this an Nth-order *moving average* model, MA(N). The origin of this (sometimes confusing) terminology can be seen if one pictures a simple smoothing filter which averages the last few values of series e. Engineers call this a *finite impulse response* (FIR) filter, because the output is guaranteed to go to zero at N time steps after the input becomes zero.

Properties of the output series x clearly depend on the input series e. The question is whether there are characteristic features independent of a specific input sequence. For a linear system, the response of the filter is independent of the input. A characterization focuses on properties of the system, rather than on properties of the time series. (For example, it does not make sense to attribute linearity to a time series itself, only to a system.)

We will give three equivalent characterizations of an MA model: in the time domain (the impulse response of the filter), in the frequency domain (its spectrum), and in terms of its autocorrelation coefficients. In the first case, we assume that the input is nonzero only at a single time step t_0 and that it vanishes for all other times. The response (in the time domain) to this "impulse" is simply given by the b's in Eq. (1): at each time step the impulse moves up to the next coefficient until, after N steps, the output disappears. The series $b_N, b_{N-1}, \cdots, b_0$ is thus the impulse response of the system. The response to an arbitrary input can be computed by superimposing the responses at appropriate delays, weighted by the respective input values ("convolution"). The transfer function thus completely describes a linear system, i.e., a system where the superposition principle holds: the output is determined by impulse response and input.

Sometimes it is more convenient to describe the filter in the frequency domain. This is useful (and simple) because a convolution in the time domain becomes a product in the frequency domain. If the input to a MA model is an impulse (which has a flat power spectrum), the discrete Fourier transform of the output is given by $\sum_{n=0}^{N} b_n \exp(-i2\pi nf)$ (see, for example, Box & Jenkins, 1976, p.69). The power spectrum is given by the squared magnitude of this:

$$\left|1 + b_1 e^{-i2\pi 1f} + b_2 e^{-i2\pi 2f} + \cdots + b_N e^{-i2\pi Nf}\right|^2 . \tag{2}$$

The third way of representing yet again the same information is, in terms of the autocorrelation coefficients, defined in terms of the mean $\mu = \langle x_t \rangle$ and the variance $\sigma^2 = \langle (x_t - \mu)^2 \rangle$ by

$$\rho_\tau \equiv \frac{1}{\sigma^2} \langle (x_t - \mu)(x_{t-\tau} - \mu) \rangle . \tag{3}$$

The angular brackets $\langle \cdot \rangle$ denote expectation values, in the statistics literature often indicated by $E\{\cdot\}$. The autocorrelation coefficients describe how much, on average, two values of a series that are τ time steps apart co-vary with each other. (We will later replace this linear measure with mutual information, suited also to describe nonlinear relations.) If the input to the system is a stochastic process with

input values at different times uncorrelated, $\langle e_i e_j \rangle = 0$ for $i \neq j$, then all of the cross terms will disappear from the expectation value in Eq. (3), and the resulting autocorrelation coefficients are

$$\rho_\tau = \begin{cases} \frac{1}{\sum_{n=0}^{N} b_n^2} \sum_{n=\tau}^{N} b_n b_{n-|\tau|} & |\tau| \leq N , \\ 0 & |\tau| > N . \end{cases} \tag{4}$$

Autoregressive (AR) models. MA (or FIR) filters operate in an open loop without feedback; they can only transform an input that is applied to them. If we do not want to drive the series externally, we need to provide some feedback (or memory) in order to generate internal dynamics:

$$x_t = \sum_{m=1}^{M} a_m x_{t-m} + e_t . \tag{5}$$

This is called an Mth-order *autoregressive model* (AR(M)) or an *infinite impulse response* (IIR) filter (because the output can continue after the input ceases). Depending on the application, e_t can represent either a controlled input to the system or noise. As before, if e is white noise, the autocorrelations of the output series x can be expressed in terms of the model coefficients. Here, however—due to the feedback coupling of previous steps—we obtain a set of linear equations rather than just a single equation for each autocorrelation coefficient. By multiplying Eq. (5) by $x_{t-\tau}$, taking expectation values, and normalizing (see Box & Jenkins, 1976, p.54), the autocorrelation coefficients of an AR model are found by solving this set of linear equations, traditionally called the *Yule-Walker equations*,

$$\rho_\tau = \sum_{m=1}^{M} a_m \rho_{\tau-m}, \qquad \tau > 0 . \tag{6}$$

Unlike the MA case, the autocorrelation coefficient need not vanish after M steps. Taking the Fourier transform of both sides of Eq. (5) and rearranging terms shows that the output equals the input times $(1 - \sum_{m=1}^{M} a_m \exp(-i2\pi mf))^{-1}$. The power spectrum of output is thus that of the input times

$$\frac{1}{\left|1 - a_1 e^{-i2\pi 1f} - a_2 e^{-i2\pi 2f} - \cdots - a_M e^{-i2\pi Mf}\right|^2} . \tag{7}$$

To generate a specific realization of the series, we must specify the initial conditions, usually by the first M values of series x. Beyond that, the input term e_t is crucial for the life of an AR model. If there was no input, we might be disappointed by the series we get: depending on the amount of feedback, after iterating

it for a while, the output produced can only decay to zero, diverge, or oscillate periodically.[7]

Clearly, the next step in complexity is to allow both AR and MA parts in the model; this is called an ARMA(M, N) model:

$$x_t = \sum_{m=1}^{M} a_m x_{t-m} + \sum_{n=0}^{N} b_n e_{t-n} \,. \tag{8}$$

Its output is most easily understood in terms of the *z-transform* (Oppenheim & Schafer, 1989), which generalizes the discrete Fourier transform to the complex plane:

$$X(z) \equiv \sum_{t=-\infty}^{\infty} x_t z^t \,. \tag{9}$$

On the unit circle, $z = \exp(-i2\pi f)$, the z-transform reduces to the discrete Fourier transform. Off the unit circle, the z-transform measures the rate of divergence or convergence of a series. Since the convolution of two series in the time domain corresponds to the multiplication of their z-transforms, the z-transform of the output of an ARMA model is

$$\begin{aligned} X(z) &= A(z)X(z) + B(z)E(z) \\ &= \frac{B(z)}{1 - A(z)} \; E(z) \end{aligned} \tag{10}$$

(ignoring a term that depends on the initial conditions). The input z-transform $E(z)$ is multiplied by a transfer function that is unrelated to it; the transfer function will vanish at zeros of the MA term ($B(z) = 0$) and diverge at poles ($A(z) = 1$) due to the AR term (unless cancelled by a zero in the numerator). As $A(z)$ is an Mth-order complex polynomial, and $B(z)$ is Nth-order, there will be M poles and N zeros. Therefore, the z-transform of a time series produced by Eq. (8) can be decomposed into a rational function and a remaining (possibly continuous) part due to the input. The number of poles and zeros determines the number of *degrees of freedom* of the system (the number of previous states that the dynamics retains). Note that since only the ratio enters, there is no unique ARMA model. In the extreme cases, a finite-order AR model can always be expressed by an infinite-order MA model, and vice versa.

ARMA models have dominated all areas of time series analysis and discrete-time signal processing for more than half a century. For example, in speech recognition and synthesis, Linear Predictive Coding (Press et al., 1992, p.571) compresses

[7] In the case of a first-order AR model, this can easily be seen: if the absolute value of the coefficient is less than unity, the value of x exponentially decays to zero; if it is larger than unity, it exponentially explodes. For higher-order AR models, the long-term behavior is determined by the locations of the zeroes of the polynomial with coefficients a_i.

speech by transmitting the slowly varying coefficients for a linear model (and possibly the remaining error between the linear forecast and the desired signal) rather than the original signal. If the model is good, it transforms the signal into a small number of coefficients plus residual white noise (of one kind or another).

3.1.2 FITTING A LINEAR MODEL TO A GIVEN TIME SERIES

Fitting the coefficients. The Yule-Walker set of linear equations (Eq. (6)) allowed us to express the autocorrelation coefficients of a time series in terms of the AR coefficients that generated it. But there is a second reading of the same equations: they also allow us to estimate the coefficients of an AR(M) model from the observed correlational structure of an observed signal.[8] An alternative approach views the estimation of the coefficients as a regression problem: expressing the next value as a function of M previous values, i.e., linearly regress x_{t+1} onto $\{x_t, x_{t-1}, \ldots, x_{t-(M-1)}\}$. This can be done by minimizing squared errors: the parameters are determined such that the squared difference between the model output and the observed value, summed over all time steps in the fitting region, is as small as possible. There is no comparable conceptually simple expression for finding MA and full ARMA coefficients from observed data. For all cases, however, standard techniques exist, often expressed as efficient recursive procedures (Box & Jenkins, 1976; Press et al., 1992).

Although there is no reason to expect that an arbitrary signal was produced by a system that can be written in the form of Eq. (8), it is reasonable to attempt to approximate a linear system's true transfer function (z-transform) by a ratio of polynomials, i.e., an ARMA model. This is a problem in function approximation, and it is well known that a suitable sequence of ratios of polynomials (called Padé approximants; see Press et al., 1992, p.200) converges faster than a power series for arbitrary functions.

Selecting the (order of the) model. So far we have dealt with the question of how to estimate the coefficients from data for an ARMA model of order (M, N), but have not addressed the choice for the order of the model. There is not a unique best choice for the values or even for the number of coefficients to model a data set—as the order of the model is increased, the fitting error decreases, but the test error of the forecasts beyond the training set will usually start to increase at some point because the model will be fitting extraneous noise in the system. There are several heuristics to find the "right" order (such as the Akaike Information Criterion (AIC), Akaike, 1970; Sakomoto et al., 1986)—but these heuristics rely heavily on the linearity of the model and on assumptions about the distribution from which the errors are drawn. When it is not clear whether these assumptions hold, a simple approach (but wasteful in terms of the data) is to hold back some of

[8] In statistics, it is common to emphasize the difference between a given model and an estimated model by using different symbols, such as $\hat{a}$ for the estimated coefficients of an AR model. In this paper, we avoid introducing another set of symbols; we hope that it is clear from the context whether values are theoretical or estimated.

the training data and use these to evaluate the performance of competing models. Model selection is a general problem that will reappear even more forcefully in the context of nonlinear models, because they are more flexible and, hence, more capable of modeling irrelevant noise.

3.2 THE BREAKDOWN OF LINEAR MODELS

We have seen that ARMA coefficients, power spectra, and autocorrelation coefficients contain the same information about a linear system that is driven by uncorrelated white noise. Thus, *if and only if* the power spectrum is a useful characterization of the relevant features of a time series, an ARMA model will be a good choice for describing it. This appealing simplicity can fail entirely for even simple nonlinearities if they lead to complicated power spectra (as they can). Two time series can have very similar broadband spectra but can be generated from systems with very different properties, such as a linear system that is driven stochastically by external noise, and a deterministic (noise-free) nonlinear system with a small number of degrees of freedom. One the key problems addressed in this chapter is how these cases can be distinguished—linear operators definitely will not be able to do the job.

Let us consider two nonlinear examples of discrete-time maps (like an AR model, but now nonlinear):

- The first example can be traced back to Ulam (1957): the next value of a series is derived from the present one by a simple parabola

$$x_{t+1} = \lambda\, x_t\, (1 - x_t)\,. \tag{11}$$

  Popularized in the context of population dynamics as an example of a "simple mathematical model with very complicated dynamics" (May, 1976), it has been found to describe a number of controlled laboratory systems such as hydrodynamic flows and chemical reactions, because of the universality of smooth unimodal maps (Collet, 1980). In this context, this parabola is called the *logistic map* or *quadratic map*. The value x_t deterministically depends on the previous value x_{t-1}; λ is a parameter that controls the qualitative behavior, ranging from a fixed point (for small values of λ) to deterministic chaos. For example, for $\lambda = 4$, each iteration destroys one bit of information. Consider that, by plotting x_t against x_{t-1}, each value of x_t has two equally likely predecessors or, equally well, the average slope (its absolute value) is two: if we know the location within ϵ before the iteration, we will on average know it within 2ϵ afterwards. This exponential increase in uncertainty is the hallmark of deterministic chaos ("divergence of nearby trajectories").
- The second example is equally simple: consider the time series generated by the map

$$x_t = 2x_{t-1} \pmod 1\,. \tag{12}$$

  The action of this map is easily understood by considering the position x_t written in a binary fractional expansion (i.e., $x_t = 0.d_1d_2\ldots = (d_1 \times 2^{-1}) + (d_2 \times 2^{-2}) + \ldots$): each iteration shifts every digit one place to the left ($d_i \leftarrow d_{i+1}$). This means that the most significant digit d_1 is discarded and one more digit of the binary expansion of the initial condition is revealed. This map can be implemented in a simple physical system consisting of a classical billiard ball and reflecting surfaces, where the x_t are the successive positions at which the ball crosses a given line (Moore, 1991).

Both systems are completely deterministic (their evolutions are entirely determined by the initial condition x_0), yet they can easily generate time series with broadband power spectra. In the context of an ARMA model a broadband component in a power spectrum of the output must come from external noise input to the system, but here it arises in two one-dimensional systems as simple as a parabola and two straight lines. Nonlinearities are essential for the production of "interesting" behavior in a deterministic system, the point here is that even simple nonlinearities suffice.

Historically, an important step beyond linear models for prediction was taken in 1980 by Tong and Lim (see also Tong, 1990). After more than five decades of approximating a system with *one* globally linear function, they suggested the use of *two* functions. This *threshold autoregressive model* (TAR) is globally nonlinear: it consists of choosing one of two local linear autoregressive models based on the value of the system's state. From here, the next step is to use many local linear models; however, the number of such regions that must be chosen may be very large if the system has even quadratic nonlinearities (such as the logistic map). A natural extension of Eq. (8) for handling this is to include quadratic and higher order powers in the model; this is called a Volterra series (Volterra, 1959).

TAR models, Volterra models, and their extensions significantly expand the scope of possible functional relationships for modeling time series, but these come at the expense of the simplicity with which linear models can be understood and fit to data. For nonlinear models to be useful, there must be a process that exploits features of the data to guide (and restrict) the construction of the model; lack of insight into this problem has limited the use of nonlinear time series models. In the next sections we will look at two complementary solutions to this problem: building explicit models with state-space reconstruction, and developing implicit models in a connectionist framework. To understand why both of these approaches exist and why they are useful, let us consider the nature of scientific modeling.

4. UNDERSTANDING AND LEARNING

Strong models have strong assumptions. They are usually expressed in a few equations with a few parameters, and can often explain a plethora of phenomena. In weak models, on the other hand, there are only a few domain-specific assumptions. To compensate for the lack of explicit knowledge, weak models usually contain many more parameters (which can make a clear interpretation difficult). It can be helpful to conceptualize models in the two-dimensional space spanned by the axes data-poor↔data-rich and theory-poor↔theory-rich. Due to the dramatic expansion of the capability for automatic data acquisition and processing, it is increasingly feasible to venture into the theory-poor and data-rich domain.

Strong models are clearly preferable, but they often originate in weak models. (However, if the behavior of an observed system does not arise from simple rules, they may not be appropriate.) Consider planetary motion (Gingerich, 1992). Tycho Brahe's (1546–1601) experimental observations of planetary motion were accurately described by Johannes Kepler's (1571–1630) phenomenological laws; this success helped lead to Isaac Newton's (1642–1727) simpler but much more general theory of gravity which could derive these laws; Henri Poincaré's (1854–1912) inability to solve the resulting three-body gravitational problem helped lead to the modern theory of dynamical systems and, ultimately, to the identification of chaotic planetary motion (Sussman & Wisdom, 1988, 1992).

As in the previous section on linear systems, there are two complementary tasks: discovering the properties of a time series generated from a given model, and inferring a model from observed data. We focus here on the latter, but there has been comparable progress for the former. Exploring the behavior of a model has become feasible in interactive computer environments, such as Cornell's `dstool`,[9] and the combination of traditional numerical algorithms with algebraic, geometric, symbolic, and artificial intelligence techniques is leading to automated platforms for exploring dynamics (Abelson, 1990; Yip, 1991; Bradley, 1992). For a nonlinear system, it is no longer possible to decompose an output into an input signal and an independent transfer function (and thereby find the correct input signal to produce a desired output), but there are adaptive techniques for controlling nonlinear systems (Hübler, 1989; Ott, Grebogi & Yorke, 1990) that make use of techniques similar to the modeling methods that we will describe.

The idea of weak modeling (data-rich and theory-poor) is by no means new—an ARMA model is a good example. What is new is the emergence of weak models (such as neural networks) that combine broad generality with insight into how to manage their complexity. For such models with broad approximation abilities and few specific assumptions, the distinction between memorization and generalization becomes important. Whereas the signal-processing community sometimes uses the term *learning* for any adaptation of parameters, we need to contrast learning without generalization from learning with generalization. Let us consider the widely and wildly celebrated fact that neural networks can learn to implement the exclusive OR (XOR). But—what kind of learning is this? When four out of four cases are specified, no generalization exists! Learning a truth table is nothing but rote memorization: learning XOR is as interesting as memorizing the phone book. More interesting—and more realistic—are real-world problems, such as the prediction of financial data. In forecasting, nobody cares how well a model fits the training data—only the quality of future predictions counts, i.e., the performance on novel data or the *generalization* ability. Learning means extracting regularities from training examples that do transfer to new examples.

Learning procedures are, in essence, statistical devices for performing inductive inference. There is a tension between two goals. The immediate goal is to fit the training examples, suggesting devices as general as possible so that they can learn a broad range of problems. In connectionism, this suggests large and flexible networks, since networks that are too small might not have the complexity needed to model the data. The ultimate goal of an inductive device is, however, its performance on cases it has not yet seen, i.e., the quality of its predictions outside the training set. This suggests—at least for noisy training data—networks that are not too large since networks with too many high-precision weights will pick out idiosyncrasies of the training set and will not generalize well.

An instructive example is polynomial curve fitting in the presence of noise. On the one hand, a polynomial of too low an order cannot capture the structure present in the data. On the other hand, a polynomial of too high an order, going through all of the training points and merely interpolating between them, captures the noise as well as the signal and is likely to be a very poor predictor for new cases. This problem of fitting the noise in addition to the signal is called *overfitting*. By employing a regularizer (i.e., a term that penalizes the complexity of the model) it is often possible to fit the parameters and to select the relevant variables at the same time. Neural networks, for example, can be cast in such a Bayesian framework (Buntine & Weigend, 1991).

To clearly separate memorization from generalization, the true continuation of the competition data was kept secret until the deadline, ensuring that the continuation data could not be used by the participants for tasks such as parameter estimation or model selection.[10] Successful forecasts of the withheld *test set* (also called *out-of-sample predictions*) from the provided *training set* (also called *fitting set*) were produced by two general classes of techniques: those based on state-space reconstruction (which make use of explicit understanding of the relationship between the internal degrees of freedom of a deterministic system and an observable of the system's state in order to build a model of the rules governing the measured behavior of the system), and connectionist modeling (which uses potentially rich

[9] Available by anonymous ftp from `macomb.tn.cornell.edu` in `pub/dstool`.

[10] After all, predictions are hard, particularly those concerning the future.

models along with learning algorithms to develop an implicit model of the system). We will see that neither is uniquely preferable. The domains of applicability are not the same, and the choice of which to use depends on the goals of the analysis (such as an understandable description vs. accurate short-term forecasts).

4.1 UNDERSTANDING: STATE-SPACE RECONSTRUCTION

Yule's original idea for forecasting was that future predictions can be improved by using immediately preceding values. An ARMA model, Eq.(8), can be rewritten as a dot product between vectors of the time-lagged variables and coefficients:

$$x_t = \mathbf{a} \cdot \mathbf{x}_{t-1} + \mathbf{b} \cdot \mathbf{e}_t \,, \tag{13}$$

where $\mathbf{x}_t = (x_t, x_{t-1}, \ldots, x_{t-(d-1)})$, and $\mathbf{a} = (a_1, a_2, \ldots, a_d)$. (We slightly change notation here: what was M (the order of the AR model) is now called d (for dimension).) Such lag vectors, also called tapped delay lines, are used routinely in the context of signal processing and time series analysis, suggesting that they are more than just a typographical convenience.[11]

In fact, there is a deep connection between time-lagged vectors and underlying dynamics. This connection was was proposed in 1980 by Ruelle (personal communication), Packard et al. (1980), and Takens (1981; he published the first proof), and later strengthened by Sauer et al. (1991). Delay vectors of sufficient length are not just a representation of the state of a linear system—it turns out that delay vectors can recover the full geometrical structure of a nonlinear system. These results address the general problem of inferring the behavior of the intrinsic degrees of freedom of a system when a function of the state of the system is measured. If the governing equations and the functional form of the observable are known in advance, then a Kalman filter is the optimal linear estimator of the state of the system (Catlin, 1989; Chatfield, 1989). We, however, focus on the case where there is little or no *a priori* information available about the origin of the time series.

There are four relevant (and easily confused) spaces and dimensions for this discussion:[12]

1. The *configuration space* of a system is the space "where the equations live." It specifies the values of all of the potentially accessible physical degrees of freedom of the system. For example, for a fluid governed by the Navier-Stokes partial differential equations, these are the infinite-dimensional degrees of freedom associated with the continuous velocity, pressure, and temperature fields.
2. The *solution manifold* is where "the solution lives," i.e., the part of the configuration space that the system actually explores as its dynamics unfolds (such as the support of an attractor or an integral surface). Due to unexcited or correlated degrees of freedom, this can be much smaller than the configuration space; the dimension of the solution manifold is the number of parameters that are needed to uniquely specify a distinguishable state of the overall system. For example, in some regimes the infinite physical degrees of freedom of a convecting fluid reduce to a small set of coupled ordinary differential equations for a mode expansion (Lorenz, 1963). Dimensionality reduction from the configuration space to the solution manifold is a common feature of dissipative systems: dissipation in a system will reduce its dynamics onto a lower dimensional subspace (Temam, 1988).
3. The *observable* is a (usually) one-dimensional function of the variables of configuration, an example is Eq. (51) in the Appendix. In an experiment, this might be the temperature or a velocity component at a point in the fluid.
4. The *reconstructed state space* is obtained from that (scalar) observable by combining past values of it to form a lag vector (which for the convection case would aim to recover the evolution of the components of the mode expansion).

Given a time series measured from such a system—and no other information about the origin of the time series—the question is: What can be deduced about the underlying dynamics?

Let $\mathbf{y}$ be the state vector on the solution manifold (in the convection example the components of $\mathbf{y}$ are the magnitude of each of the relevant modes), let $d\mathbf{y}/dt = \mathbf{f}(\mathbf{y})$ be the governing equations, and let the measured quantity be $x_t = x(\mathbf{y}(t))$ (e.g., the temperature at a point). The results to be cited here also apply to systems that are described by iterated maps. Given a delay time τ and a dimension d, a lag vector $\mathbf{x}$ can be defined,

$$\text{lag vector}: \quad \mathbf{x}_t = (x_t, x_{t-\tau}, \ldots, x_{t-(d-1)\tau}) \,. \tag{14}$$

The central result is that the behavior of $\mathbf{x}$ and $\mathbf{y}$ will differ only by a smooth local invertible change of coordinates (i.e., the mapping between $\mathbf{x}$ and $\mathbf{y}$ is an embedding, which requires that it be diffeomorphic) for almost every possible choice of $\mathbf{f}(\mathbf{y})$, $x(\mathbf{y})$, and τ, as long as d is large enough (in a way that we will make precise), x depends on at least some of the components of $\mathbf{y}$, and the remaining components of $\mathbf{y}$ are coupled by the governing equations to the ones that influence $\mathbf{x}$. The proof of this result has two parts: a local piece, showing that the linearization of the embedding map is almost always nondegenerate, and a global part, showing

[11] For example, the spectral test for random number generators is based on looking for structure in the space of lagged vectors of the output of the source; these will lie on hyperplanes for a linear congruential generator $x_{t+1} = ax_t + b \pmod{c}$ (Knuth, 1981, p.90).

[12] The first point (configuration space and potentially accessible degrees of freedom) will not be used again in this chapter. On the other hand, the dimension of the solution manifold (the actual degrees of freedom) will be important both for characterization and for prediction.

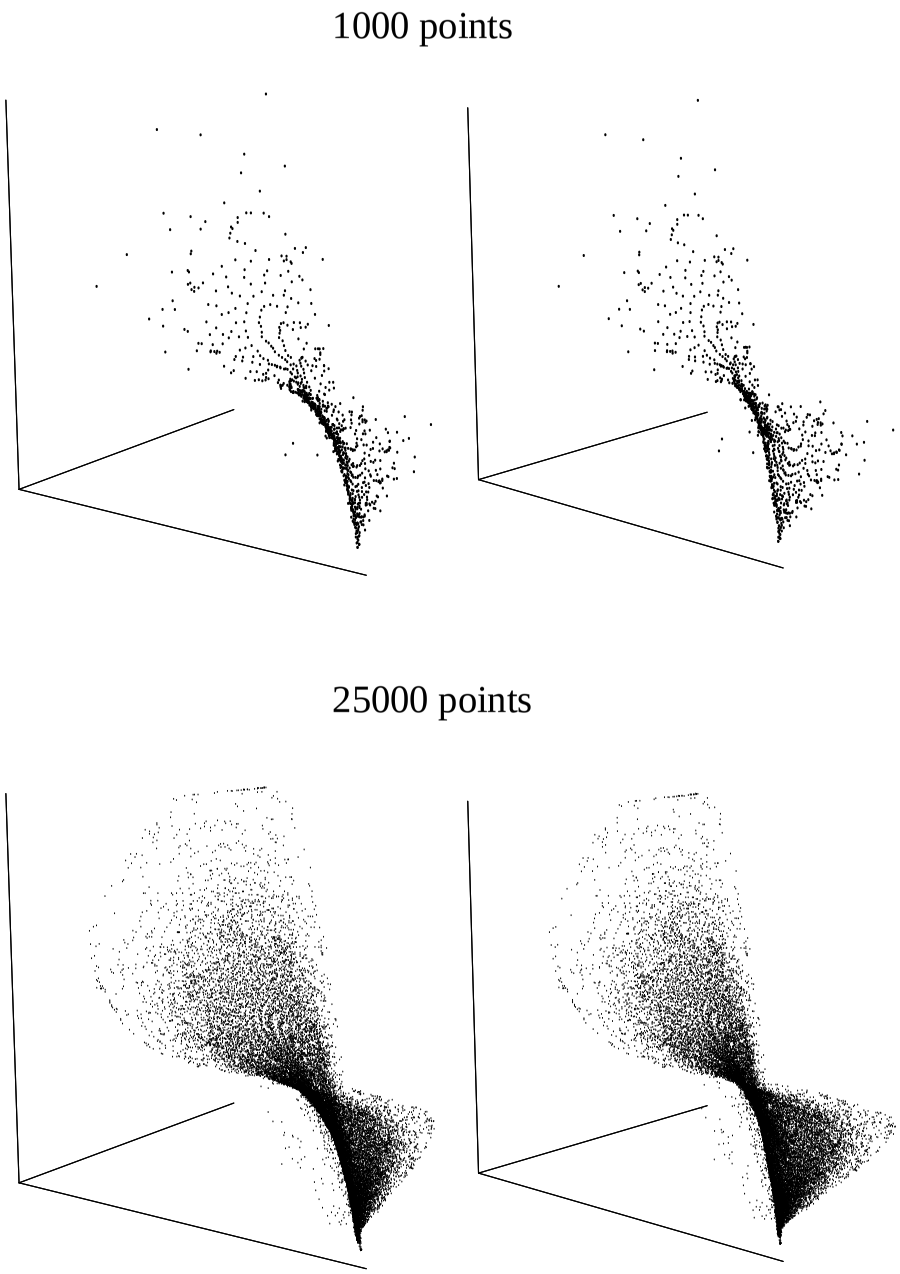

FIGURE 5 Stereo pairs for the three-dimensional embedding of Data Set A. The shape of the surface is apparent with just the 1,000 points that were given.

that this holds everywhere. If τ tends to zero, the embedding will tend to lie on the diagonal of the embedding space and, as τ is increased, it sets a length scale for the reconstructed dynamics. There can be degenerate choices for τ for which the embedding fails (such as choosing it to be exactly equal to the period of a periodic system), but these degeneracies almost always will be removed by an arbitrary perturbation of τ. The intrinsic noise in physical systems guarantees that these results hold in all known nontrivial examples, although in practice, if the coupling between degrees of freedom is sufficiently weak, then the available experimental resolution will not be large enough to detect them (see Casdagli et al., 1991, for further discussion of how noise constrains embedding).[13]

Data Set A appears complicated when plotted as a time series (Figure 1). The simple structure of the system becomes visible in a figure of its three-dimensional embedding (Figure 5). In contrast, high-dimensional dynamics would show up as a structureless cloud in such a stereo plot. Simply plotting the data in a stereo plot allows one to guess a value of the dimension of the manifold of around two, not far from computed values of 2.0-2.2. In Section 6, we will discuss in detail the practical issues associated with choosing and understanding the embedding parameters.

Time-delay embedding differs from traditional experimental measurements in three fundamental respects:

1. It provides detailed information about the behavior of degrees of freedom other than the one that is directly measured.
2. It rests on probabilistic assumptions and—although it has been routinely and reliably used in practice—it is not guaranteed to be valid for any system.
3. It allows precise questions only about quantities that are invariant under such a transformation, since the reconstructed dynamics have been modified by an unknown smooth change of coordinates.

This last restriction may be unfamiliar, but it is surprisingly unimportant: we will show how embedded data can be used for forecasting a time series and for characterizing the essential features of the dynamics that produced it. We close this section by presenting two extensions of the simple embedding considered so far.

Filtered embedding generalizes simple time-delay embedding by presenting a linearly transformed version of the lag vector to the next processing stage. The lag vector $\mathbf{x}$ is trivially equal to itself times an identity matrix. Rather than using the identity matrix, the lag vector can be multiplied by any (not necessarily square) matrix. The resulting vector is an embedding if the rank of the matrix is equal to or larger than the desired embedding dimension. (The window of lags can be larger than the final embedding dimension, which allows the embedding procedure

[13] The Whitney embedding theorem from the 1930s (see Guillemin & Pollack, 1974, p. 48) guarantees that the number of independent observations d required to embed an arbitrary manifold (in the absence of noise) into a Euclidean embedding space will be no more than twice the dimension of the manifold. For example, a two-dimensional Möbius strip can be embedded in a three-dimensional Euclidean space, but a two-dimensional Klein bottle requires a four-dimensional space.

to include additional signal processing.) A specific example, used by Sauer (this volume), is embedding with a matrix produced by multiplying a discrete Fourier transform, a low-pass filter, and an inverse Fourier transform; as long as the filter cut-off is chosen high enough to keep the rank of the overall transformation greater than or equal to the required embedding dimension, this will remove noise but will preserve the embedding. There are a number of more sophisticated linear filters that can be used for embedding (Oppenheim & Schafer, 1989), and we will also see that connectionist networks can be interpreted as sets of nonlinear filters.

A final modification of time-delay embedding that can be useful in practice is **embedding by expectation values.** Often the goal of an analysis is to recover not the detailed trajectory of $\mathbf{x}(t)$, but rather to estimate the probability distribution $p(\mathbf{x})$ for finding the system in the neighborhood of a point $\mathbf{x}$. This probability is defined over a measurement of duration T in terms of an arbitrary test function $g(\mathbf{x})$ by

$$\begin{aligned} \frac{1}{T}\int_0^T g(\mathbf{x}(t))dt &= \langle g(\mathbf{x}(t))\rangle_t \\ &= \int g(\mathbf{x})p(\mathbf{x})\ d\mathbf{x}\,. \end{aligned} \tag{15}$$

Note that this is an empirical definition of the probability distribution for the observed trajectory; it is not equivalent to assuming the existence of an invariant measure or of ergodicity so that the distribution is valid for all possible trajectories (Petersen, 1989). If a complex exponential is chosen for the test function

$$\begin{aligned} \langle \mathrm{e}^{i\mathbf{k}\cdot\mathbf{x}(t)}\rangle &= \langle \mathrm{e}^{i\mathbf{k}\cdot(x_t, x_{t-\tau},\ldots,x_{t-(d-1)\tau})}\rangle \\ &= \int \mathrm{e}^{i\mathbf{k}\cdot\mathbf{x}} p(\mathbf{x})\ d\mathbf{x}\,, \end{aligned} \tag{16}$$

we see that the time average of this is equal to the Fourier transform of the desired probability distribution (this is just a characteristic function of the lag vector). This means that, if it is not possible to measure a time series directly (such as for very fast dynamics), it can still be possible to do time-delay embedding by measuring a set of time-average expectation values and then taking the inverse Fourier transform to find $p(\mathbf{x})$ (Gershenfeld, 1993a). We will return to this point in Section 6.2 and show how embedding by expectation values can also provide a useful framework for distinguishing measurement noise from underlying dynamics.

We have seen that time-delay embedding, while appearing similar to traditional state-space models with lagged vectors, makes a crucial link between behavior in the reconstructed state space and the internal degrees of freedom. We will apply this insight to forecasting and characterizing deterministic systems later (in Sections 5.1 and 6.2). Now, we address the problem of what can be done if we are unable to understand the system in such explicit terms. The main idea will be to learn to emulate the behavior of the system.

4.2 LEARNING: NEURAL NETWORKS

In the competition, the majority of contributions, and also the best predictions for each set used connectionist methods. They provide a convenient language game for nonlinear modeling. Connectionist networks are also known as neural networks, parallel distributed processing, or even as "brain-style computation"; we use these terms interchangeably. Their practical application (such as by large financial institutions for forecasting) has been marked by (and marketed with) great hope and hype (Schwarz, 1992; Hammerstrom, 1993).

Neural networks are typically used in pattern recognition, where a collection of features (such as an image) is presented to the network, and the task is to assign the input feature to one or more classes. Another typical use for neural networks is (nonlinear) regression, where the task is to find a smooth interpolation between points. In both these cases, all the relevant information is presented simultaneously. In contrast, time series prediction involves processing of patterns that evolve over time—the appropriate response at a particular point in time depends not only on the current value of the observable but also on the past. Time series prediction has had an appeal for neural networkers from the very beginning of the field. In 1964, Hu applied Widrow's adaptive linear network to weather forecasting. In the post-backpropagation era, Lapedes and Farber (1987) trained their (nonlinear) network to emulate the relationship between output (the next point in the series) and inputs (its predecessors) for computer-generated time series, and Weigend, Huberman and Rumelhart (1990, 1992) addressed the issue of finding networks of appropriate complexity for predicting observed (real-world) time series. In all these cases, temporal information is presented spatially to the network by a time-lagged vector (also called tapped delay line).

A number of ingredients are needed to specify a neural network:

- its interconnection architecture,
- its activation functions (that relate the output value of a node to its inputs),
- the cost function that evaluates the network's output (such as squared error),
- a training algorithm that changes the interconnection parameters (called weights) in order to minimize the cost function.

The simplest case is a network **without hidden units**: it consists of one output unit that computes a weighted linear superposition of d inputs, $\mathtt{out}^{(t)} = \sum_{i=1}^{d} w_i x_i^{(t)}$. The superscript $^{(t)}$ denotes a specific "pattern"; $x_i^{(t)}$ is the value of the ith input of that pattern.[14] w_i is the weight between input i and the output. The network output can also be interpreted as a dot-product $\mathbf{w}\cdot\mathbf{x}^{(t)}$ between the weight vector $\mathbf{w} = (w_1, \cdots, w_d)$ and an input pattern $\mathbf{x}^{(t)} = (x_1^{(t)}, \ldots, x_d^{(t)})$.

[14] In the context of time series prediction, $x_i^{(t)}$ can be the ith component of the delay vector, $x_i^{(t)} = x_{t-i}$.

Given such an input-output relationship, the central task in learning is to find a way to change the weights such that the actual output $\texttt{out}^{(t)}$ gets closer to the desired output or $\texttt{target}^{(t)}$. The closeness is expressed by a cost function, for example, the squared error $E^{(t)} = (\texttt{out}^{(t)} - \texttt{target}^{(t)})^2$. A learning algorithm iteratively updates the weights by taking a small step (parametrized by the learning rate η) in the direction that decreases the error the most, i.e., following the negative of the local gradient.[15] The "new" weight $\widetilde{w}_i$, after the update, is expressed in terms of the "old" weight w_i as

$$\widetilde{w}_i = w_i - \eta \frac{\partial E^{(t)}}{\partial w_i} = w_i + \underbrace{2\eta \; x_i}_{\text{activation}} \underbrace{(\texttt{out}^{(t)} - \texttt{target}^{(t)})}_{\text{error}} . \tag{17}$$

The weight-change $(\widetilde{w}_i - w_i)$ is proportional to the product of the activation going into the weight and the size of error, here the deviation $(\texttt{out}^{(t)} - \texttt{target}^{(t)})$. This rule for adapting weights (for linear output units with squared errors) goes back to Widrow and Hoff (1960).

If the input values are the lagged values of a time series and the output is the prediction for the next value, this simple network is equivalent to determining an AR(d) model through least squares regression: the weights at the end of training equal the coefficients of the AR model.

Linear networks are very limited—exactly as limited as linear AR models. The key idea responsible for the power, potential, and popularity of connectionism is the insertion of one of more layers of **nonlinear hidden units** (between the inputs and output). These nonlinearities allow for interactions between the inputs (such as products between input variables) and thereby allow the network to fit more complicated functions. (This is discussed further in the subsection on neural networks and statistics below.)

The simplest such nonlinear network contains only one hidden layer and is defined by the following components:

- There are d *inputs*.
- The inputs are fully connected to a layer of nonlinear *hidden units*.
- The hidden units are connected to the one linear *output unit*.
- The output and hidden units have adjustable offsets or *biases* b.
- The *weights* w can be positive, negative, or zero.

The response of a unit is called its activation value or, in short, *activation*. A common choice for the nonlinear *activation function* of the hidden units is a

[15] Eq. (17) assumes updates after each pattern. This is a stochastic approximation (also called "on-line updates" or "pattern mode") to first averaging the errors over the entire training set, $E = 1/N \sum_t E^{(t)}$ and then updating (also called "batch updates" or "epoch mode"). If there is repetition in the training set, learning with pattern updates is faster.

composition of two operators: an affine mapping followed by a sigmoidal function. First, the inputs into a hidden unit h are linearly combined and a bias b_h is added:

$$\xi_h^{(t)} = \sum_{i=1}^{d} w_{hi} x_i^{(t)} + b_h \, . \tag{18}$$

Then, the output of the unit is determined by passing $\xi_h^{(t)}$ through a sigmoidal function ("squashing function") such as

$$S(\xi_h^{(t)}) = \frac{1}{1 + e^{-a\xi_h^{(t)}}} = \frac{1}{2}\left(1 + \tanh \frac{a}{2}\xi_h^{(t)}\right) , \tag{19}$$

where the slope a determines the steepness of the response.

In the introductory example of a linear network, we have seen how to change the weights when the activations are known at both ends of the weight. How do we update the weights to the hidden units that do not have a target value? The revolutionary (but in hindsight obvious) idea that solved this problem is the chain rule of differentiation. This idea of **error backpropagation** can be traced back to Werbos (1974), but only found widespread use after it was independently invented by Rumelhart et al. (1986a, 1986b) at a time when computers had become sufficiently powerful to permit easy exploration and successful application of the backpropagation rule.

As in the linear case, weights are adjusted by taking small steps in the direction of the negative gradient, $-\partial E/\partial w$. The weight-change rule is still of the same form (activation into the weight) $\times$ (error signal from above). The activation that goes into the weight remains unmodified (it is the same as in the linear case). The difference lies in the error signal. For weights between hidden unit h and the output, the error signal for a given pattern is now $(\texttt{out}^{(t)} - \texttt{target}^{(t)}) \times S'(\xi_h^{(t)})$; i.e., the previous difference between prediction and target is now multiplied with the derivative of the hidden unit activation function taken at $\xi_h^{(t)}$. For weights that do not connect directly to the output, the error signal is computed recursively in terms of the error signals of the units to which it directly connects, and the weights of those connections. The weight change is computed locally from incoming activation, the derivative, and error terms from above multiplied with the corresponding weights.[16] Clear derivations of backpropagation can be found in Rumelhart, Hinton, and Williams (1986a) and in the textbooks by Hertz, Krogh, and Palmer (1991, p.117) and Kung (1993, p.154). The theoretical foundations of backpropagation are laid out clearly by Rumelhart et al. (1993).

[16] All update rules are local. The flip side of the locality of the update rules discussed above is that learning becomes an iterative process. Statisticians usually do not focus on the emergence of iterative local solutions.

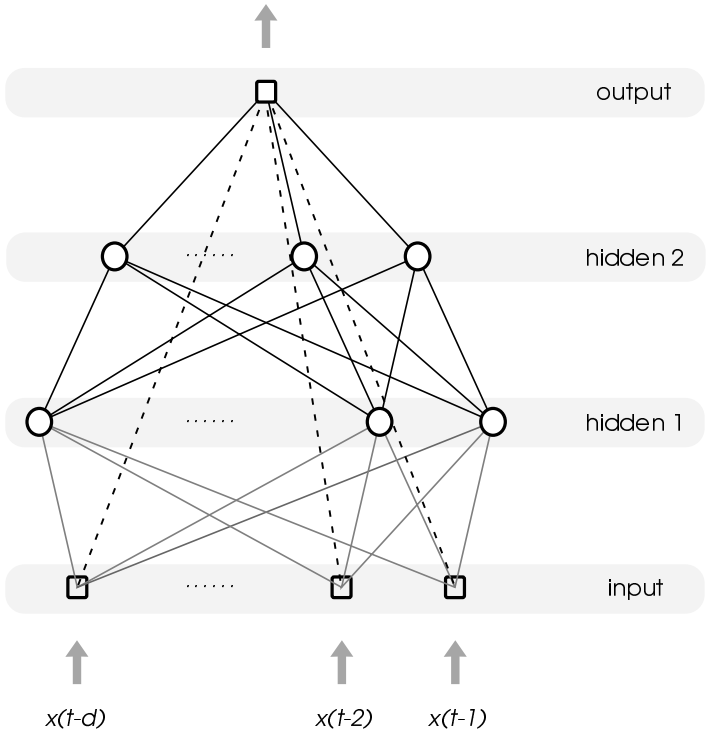

FIGURE 6 Architecture of a feedforward network with two hidden layers and direct connections from the input to the output. The lines correspond to weight values. The dashed lines represent direct connections from the inputs to the (linear) output unit. Biases are not shown.

Figure 6 shows a typical network; activations flow from the bottom up. In addition to a second layer of (nonlinear) hidden units, we also include direct (linear) connections between each input and the output. Although not used by the competition entrants, this architecture can extract the linearly predictable part early in the learning process and free up the nonlinear resources to be employed where they are really needed. It can be advantageous to choose different learning rates for different parts of the architecture, and thus not follow the gradient exactly (Weigend, 1991). In Section 6.3.2 we will describe yet another modification, i.e., sandwiching a bottleneck hidden layer between two additional (larger) layers.

NEURAL NETWORKS AND STATISTICS. Given that feedforward networks with hidden units implement a nonlinear regression of the output onto the inputs, what features do they have that might give them an advantage over more traditional methods? Consider polynomial regression. Here the components of the input vector $(x_1, x_2, \ldots, x_d)$ can be combined in pairs $(x_1x_2, x_1x_3, \ldots)$, in triples $(x_1x_2x_3, x_1x_2x_4, \ldots)$, etc., as well as in combinations of higher powers. This vast number of possible terms can approximate any desired output surface. One might be tempted to conjecture that feedforward networks are able to represent a larger function space with fewer parameters. This, however, is not true: Cover (1965) and Mitchison and Durbin (1989) showed that the "capacity" of both polynomial expansions and networks is proportional to the number of parameters. The real difference between the two representations is in the kinds of constraints they impose. For the polynomial case, the number of possible terms grows rapidly with the input dimension, making it sometimes impossible to use even all of the second-order terms. Thus, the necessary selection of which terms to include implies a decision to permit only specific pairwise or perhaps three-way interactions between components of the input vector. A layered network, rather than limiting the *order* of the interactions, limits only the total *number* of interactions and learns to select an appropriate combination of inputs. Finding a simple representation for a complex signal might require looking for such simultaneous relationships among many input variables. A small network is already potentially fully nonlinear. Units are added to increase the number of features that can be represented (rather than to increase the model order in the example of polynomial regression).

NEURAL NETWORKS AND MACHINE LEARNING. Theoretical work in connectionism ranges from reassuring proofs that neural networks with sigmoid hidden units can essentially fit any well-behaved function and its derivative (Irie & Miyake, 1988; Cybenko, 1989; Funahashi, 1989; White, 1990; Barron, 1993) to results on the ability to generalize (Haussler, personal communication, 1993).[17] Neural networks have found their place in (and helped develop) the broader field of machine learning which studies algorithms for learning from examples. For time series prediction, this has included genetic algorithms (Packard, 1990; Meyer & Packard, 1992; see also Koza, 1993, for a recent monograph on genetic programming), Boltzmann machines (Hinton & Sejnowski, 1986), and conventional AI techniques (Laird & Saul, 1993). The increasing number of such techniques that arrive with strong claims about their performance is forcing the machine learning community to pay greater attention to methodological issues. In this sense, the comparative evaluations of the Santa Fe Competition can also be viewed as a small stone in the mosaic of machine learning.

[17] This paper by Haussler (on PAC [probably approximately correct] learning) will be published in the collection edited by Smolensky, Mozer, and Rumelhart (1994). That collection also contains theoretical connectionist results by Vapnik (on induction principles), Judd (on complexity of learning), and Rissanen (on information theory and neural networks).

5. FORECASTING

In the previous section we have seen that analysis of the geometry of the embedded data and machine learning techniques provide alternative approaches to discovering the relationship between past and future points in a time series. Such insight can be used to forecast the unknown continuation of a given time series. In this section we will consider the details of how prediction is implemented, and in the following section we will step back to look at the related problem of characterizing the essential properties of a system.

5.1 STATE-SPACE FORECASTING

If an experimentally observed quantity arises from deterministic governing equations, it is possible to use time-delay embedding to recover a representation of the relevant internal degrees of freedom of the system from the observable. Although the precise values of these reconstructed variables are not meaningful (because of the unknown change of coordinates), they can be used to make precise forecasts because the embedding map preserves their geometrical structure. In this section we explain how this is done for a time series that has been generated by a deterministic system; in Section 6.2 we will consider how to determine whether or not this is the case (and, if so, what the embedding parameters should be) and, in Section 5.2, how to forecast systems that are not simply deterministic.

Figure 5 is an example of the structure that an embedding can reveal. Notice that the surface appears to be single-valued; this, in fact, must be the case if the system is deterministic and if the number of time lags used is sufficient for an embedding. Differential equations and maps have unique solutions forward in time; this property is preserved under a diffeomorphic transformation and so the first component of an embedded vector must be a unique function of the preceding values $x_{t-\tau}, ..., x_{t-(d-1)\tau}$ once d is large enough. Therefore, the points must lie on a single-valued hypersurface. Future values of the observable can be read off from this surface if it can be adequately estimated from the given data set (which may contain noise and is limited in length).

Using embedding for forecasting appears—at first sight—to be very similar to Yule's original AR model: a prediction function is sought based on time-lagged vectors. The crucial difference is that understanding embedding reduces forecasting to recognizing and then representing the underlying geometrical structure, and once the number of lags exceeds the minimum embedding dimension, this geometry will not change. A global linear model (AR) must do this with a single hyperplane. Since this may be a very poor approximation, there is no fundamental insight into how to choose the number of delays and related parameters. Instead, heuristic rules

such as the AIC[18] are used (and vigorously debated). The ease of producing and exploring pictures, such as Figure 5, with modern computers has helped clarify the importance of this point.

Early efforts to improve global linear AR models included systematically increasing the order of interaction ("bilinear" models[19]; Granger & Anderson, 1978), splitting the input space across one variable and allowing for two AR models (threshold autoregressive models, Tong & Lim, 1980), and using the nonlinearities of a Volterra expansion. A more recent example of the evolutionary improvements are adaptive fits with local splines (Multivariate Adaptive Regression Splines, MARS; Friedman, 1991c; Lewis et al., this volume). The "insight" gained from a model such as MARS describes what parameter values are used for particular regions of state space, but it does not help with deeper questions about the nature of a system (such as how many degrees of freedom there are, or how trajectories evolve). Compare this to forecasting based on state-space embedding, which starts by testing for the presence of identifiable geometrical structure and then proceeds to model the geometry, rather than starting with (often inadequate) assumptions about the geometry. This characterization step (to be discussed in detail in Section 6.2) is crucial: simple state-space forecasting becomes problematic if there is a large amount of noise in the system, or if there are nonstationarities on the time scale of the sampling time. Assuming that sufficiently low-dimensional dynamics has been detected, the next step is to build a model of the geometry of the hypersurface in the embedding space that can interpolate between measured points and can distinguish between measurement noise and intrinsic dynamics. This can be done by both local and global representations (as well as by intermediate hybrids).

Farmer and Sidorowich (1987) introduced *local linear models* for state-space forecasting. The simple idea is to recognize that any manifold is locally linear (i.e., locally a hyperplane). Furthermore, the constraint that the surface must be single-valued allows noise transverse to the surface (the generic case) to be recognized and eliminated. Broomhead and King (1986) use *Singular Value Decomposition* (SVD) for this projection; the distinction between local and global SVD is crucial. (Fraser, 1989a, points out some problems with the use of SVD for nonlinear systems.) In this volume, Smith discusses the relationship between local linear and nonlinear models,

[18] For linear regression, it is sometimes possible to "correct" for the usually over-optimistic estimate. An example is to multiply the fitting error with $(N+k)/(N-k)$, where N is the number of data points and k is the number of parameters of the model (Akaike, 1970; Sakomoto et al., 1986). Moody (1992) extended this for nonlinear regression and used a notion of effective number of parameters for a network that has convergenced. Weigend and Rumelhart (1991a, 1991b) focused on the increase of the effective network size (expressed as the effective number of hidden units) as a function on training time.

[19] A bilinear model contains second-order interactions between the inputs, i.e., $x_i x_j$. The term "bilinear" comes from the fact that two inputs enter linearly into such products.

as well as the relationship between local and global approaches. Finally, Casdagli and Weigend (this volume) specifically explore the continuum between local and global models by varying the size of the local neighborhood used in the local linear fit. Before returning to this issue in Section 6.3 where we will show how this variation relates to (and characterizes) a system's properties, we now summarize the method used by Tim Sauer in his successful entry. (Details are given by Sauer, this volume.)

In his competition entry, shown in Figure 3, Sauer used a careful implementation of local-linear fitting that had five steps:

1. Low-pass embed the data to help remove measurement and quantization noise. This low-pass filtering produces a smoothed version of the original series. (We explained such filtered embedding at the end of Section 4.1.)
2. Generate more points in embedding space by (Fourier-) interpolating between the points obtained from Step 1. This is to increase the coverage in embedding space.
3. Find the k nearest neighbors to the point of prediction (the choice of k tries to balance the increasing bias and decreasing variance that come from using a larger neighborhood).
4. Use a local SVD to project (possibly very noisy) points onto the local surface. (Even if a point is very far away from the surface, this step forces the dynamics back on the reconstructed solution manifold.)
5. Regress a linear model for the neighborhood and use it to generate the forecast.

Because Data Set A was generated by low-dimensional smooth dynamics, such a local linear model is able to capture the geometry remarkably well based on the relatively small sample size. The great advantage of local models is their ability to adhere to the local shape of an arbitrary surface; the corresponding disadvantage is that they do not lead to a compact description of the system. Global expansions of the surface reverse this tradeoff by providing a more manageable representation at the risk of larger local errors. Giona et al. (1991) give a particularly nice approach to global modeling that builds an orthogonal set of basis functions with respect to the natural measure of the attractor rather than picking a fixed set independent of the data. If $x_{t+1} = f(\mathbf{x}_t)$, $\rho(\mathbf{x})$ is the probability distribution for the state vector $\mathbf{x}$, and $\{p_i\}$ denotes a set of polynomials that are orthogonal with respect to this distribution:

$$\langle p_i(\mathbf{x}) p_j(\mathbf{x}) \rangle = \int p_i(\mathbf{x}) p_j(\mathbf{x}) \rho(\mathbf{x}) \, d\mathbf{x} = \delta_{ij} \, , \tag{20}$$

then the expansion coefficients

$$f(\mathbf{x}) = \sum_i a_i p_i(\mathbf{x}) \tag{21}$$

can be found from the time average by the orthogonality condition:

$$a_i = \langle f(\mathbf{x}_t) p_i(\mathbf{x}_t) \rangle = \langle x_{t+1} p_i(\mathbf{x}_t) \rangle = \lim_{N \to \infty} \frac{1}{N} \sum_{t=1}^{N} x_{t+1} p_i(\mathbf{x}_t) \, . \tag{22}$$

The orthogonal polynomials can be found from Gram-Schmidt orthogonalization on the moments of the time series. This expansion is similar in spirit to embedding by expectation values presented in Eq. (16).

In between global and local models lie descriptions such as **radial basis functions** (Powell, 1987; Broomhead & Lowe, 1988; Casdagli, 1989; Poggio & Girosi, 1990; L. A. Smith, this volume). A typical choice is a mixture of (spherically symmetric) Gaussians, defined by

$$f(\mathbf{x}) = \sum_i w_i e^{-(\mathbf{x} - \mathbf{c}_i)^2 / (2\sigma_i^2)} \, . \tag{23}$$

For each basis function, three quantities have to be determined: its center, $\mathbf{c}_i$; its width, σ_i; and the weight, w_i. In the simplest case, all the widths and centers are fixed. (The centers are, for example, placed on the observed data points.) In a weaker model, these assumptions are relaxed: the widths can be made adaptive, the constraint of spherical Gaussians can be removed by allowing for a general covariance matrix, and the centers can be allowed to adapt freely.

An important issue in function approximation is whether the adjustable parameters are all "after" the nonlinearities (for radial basis functions this corresponds to fixed centers and widths), or whether some of them are also "before" the nonlinearities (i.e., the centers and/or widths can be adjusted). The advantage of the former case is that the only remaining free parameters, the weights, can be estimated by matrix inversion. Its disadvantage is an exponential "curse of dimensionality."[20] In the latter case of adaptive nonlinearities, parameter estimation is harder, but can always be cast in an error backpropagation framework, i.e., solved with gradient descent. The surprising—and promising—result is that in this case when the adaptive parameters are "before" the nonlinearity, the curse of dimensionality is only linear with the dimension of the input space (Barron, 1993).

A goal beyond modeling the geometry of the manifold is to find a set of differential equations that might have produced the time series. This is often a more compact (and meaningful) description, as shown for simple examples by Cremers and Hübler (1987), and by Crutchfield and McNamara (1987). An alternative goal, trying to characterize symbol sequences (which might be obtained by a coarse quantization of real-valued data where each bin has a symbol assigned to it), is suggested by Crutchfield and Young (1989), who try to extract the rules of an automaton that could have generated the observed symbol sequence. Such approaches are interesting but are not yet routinely applicable.

[20] The higher the dimension of a data set of a given size, the more sparse the data set appears. If the average distance ϵ to the nearest point is to remain constant, the number of points N needed to cover the space increases exponentially with the dimension of the space d, $N \propto (1/\epsilon)^d$.

5.2 CONNECTIONIST FORECASTING

State-space embedding "solves" the forecasting problem for a low-dimensional deterministic system. If there is understandable structure in the embedding space, it can be detected and modeled; the open questions have to do with finding good representations of the surface and with estimating the reliability of the forecast. This approach of reconstructing the geometry of the manifold will fail if the system is high-dimensional, has stochastic inputs, or is nonstationary, because in these cases, there is no longer a simple surface to model.

Neural networks do not build an explicit description of a surface. On the one hand, this makes it harder to interpret them even for simple forecasting tasks. On the other hand, they promise to be applicable (and mis-applicable) to situations where simpler, more explicit approaches fail: much of their promise comes from the hope that they can learn to emulate unanticipated regularities in a complex signal. This broad attraction leads to results like those seen in Table 2 in the Appendix: the best as well as many of the worst forecasts of Data Set A were obtained with neural networks. The purpose of this section is to examine how neural networks can be used to forecast relatively simple time series (Data Set A, laser) as well as more difficult time series (Data Set D, high-dimensional chaos; Data Set E, currency exchange rates). We will start with *point predictions*, i.e., predictions of a single value at each time step (such as the most likely one), and then turn to the additional estimation of error bars, and eventually to the prediction of the full probability distribution.

A network that is to predict the future must know about the past. The simplest approach is to provide time-delayed samples to its input layer. In Section 4.2 we discussed that, on the one hand, a network without (nonlinear) hidden units is equivalent to an AR model (one linear filter). On the other hand, we showed that with nonliner hidden units, the network combines a number of "squashed" filters.

Eric Wan (at the time a graduate student at Stanford) used a somewhat more difficult architecture for his competition entry. The key modification to the network displayed in Figure 6 is that each connection now becomes an AR filter (tapped delay line). Rather than displaying the explicit buffer of the input units, it suffices then to draw only a single input unit and conceptually move the weight vectors into the tapped delay lines from the input to each hidden unit. This architecture is also known as "time-delay neural network" by Lang, Waibel, and Hinton (1990) or (in the spatial domain) as a network with "linked weights," as suggested by le Cun (1989).

We would like to emphasize that these architectures are all examples of *feed-forward* networks: they are trained in "open loop," and there is no feedback of activations in training. (For iterated predictions, however, predictions must be used for the input: the network is run in "closed loop" mode. A more consistent scheme is to train the network in the mode eventually used in prediction; we return to this point at the end of this section.)

Wan's network had 1 input unit, two layers of 12 hidden units each, and 1 output unit. The "generalized weights" of the first layer were tapped delay lines with 25 taps; the second and third layers had 5 taps each. These values are not the result of a simple quantitative analysis, but rather the result of evaluating the performance of a variety of architectures on some part of the available data that Wan had set aside for this purpose. Such careful exploration is important for the successful use of neural networks.

At first sight, selecting an architecture with 1,105 parameters to fit 1,000 data points seems absurd. How is it possible not to overfit if there are more parameters than data points? The key is knowing when to stop. At the outset of training, the parameters have random values, and so changing any one coefficient has little impact on the quality of the predictions. As training progresses and the fit improves, the *effective number of parameters* grows (Weigend & Rumelhart, 1991a, 1991b). The overall error in predicting points out of the training set will initially decrease as the network learns to do something, but then will begin to increase once the network learns to do too much; the location of the minimum of the "cross validation" error determines when the effective network complexity is right.[21]

The competition entry was obtained by a network that was stopped early. It did very well over the prediction interval (Figure 3), but notice how it fails dramatically after 100 time steps (Figure 4) while the local linear forecast continues on. The reason for this difference is that the local linear model with local SVD (described in Section 5.1) is constrained to stay on the reconstructed surface in the embedding space (any point gets projected onto the reconstructed manifold before the prediction is made), whereas the network does not have a comparable constraint.

Short-term predictors optimize the parameters for forecasting the next step. The same architecture (e.g., Figure 6) can also be used to follow the trajectory more closely in the longer run, at the expense of possibly worse single-step predictions, by using the *predicted* value at the input in training, rather than the *true* value.[22] There is a continuum between these two extremes: in training, the errors from both sources can be combined in a weighted way, e.g., $\lambda \times$ (short-term error) + $(1 - \lambda) \times$ (long-term error). This mixed error is then used in backpropagation. Since the network produces very poor predictions at the beginning of the training process, it can be advantageous to begin with $\lambda = 1$ (full teacher forcing) and then anneal to the desired value.

[21] Other ideas, besides early stopping, for arriving at networks of an appropriate size include (i) penalizing network complexity by adding a term to the error function, (ii) removing unimportant weights with the help of the Hessian (the second derivative of the error with respect to the weights), and (iii) expressing the problem in a Bayesian framework.

[22] There are a number of terms associated with this distinction. Engineers use the expression "open loop" when the inputs are set to true values, and "closed loop" for the case when the input are given the predicted values. In the connectionist community, the term "teacher forcing" is used when the inputs are set to the true values, and the term "trajectory learning" (Principe et al., 1993) when the predicted output is fed back to the inputs.

We close the section on point-predictions with some remarks on the question whether predictions for several steps in the future should be made by iterating the single-step prediction T times or by making a noniterated, direct prediction. Farmer and Sidorowich (1988) argue that for deterministic chaotic systems, iterated forecasts lead to better predictions than direct forecasts.[23] However, for noisy series the question "iterated vs. direct forecasts?" remains open. The answer depends on issues such as the sampling time (if the sampling is faster than the highest frequency in the system, one-step predictions will focus on the noise), and the complexity of the input-output map (even the simple parabola of the logistic map becomes a polynomial of order 2^T when direct T step predictions are attempted). As Sauer (this volume) points out, the dichotomy between iterated and direct forecasts is not necessary: combining both approaches leads to better forecasts than using either individually.

5.3 BEYOND POINT-PREDICTIONS

So far we have discussed how to predict the continuation of a time series. It is often desirable and important to also know the confidence associated with a prediction. Before giving some suggestions, we make explicit the two assumptions that minimizing sum squared errors implies in a maximum likelihood framework (which we need not accept):

- The errors of different data points are independent of each other: statistical independence is present if the joint probability between two events is precisely the product between the two individual probabilities. After taking the logarithm, the product becomes a sum. Summing errors thus assumes statistical independence of the measurement errors.
- The errors are Gaussian distributed: i.e., the likelihood of a data point given the prediction is a Gaussian. Taking the logarithm of a Gaussian transforms it into a squared difference. This squared difference can be interpreted as a squared error. Summing squared errors with the same weight for each data point assumes that the uncertainty is the same for each data point (i.e., that the size of the error bar is independent of the location in state space).

The second assumption is clearly violated in the laser data: the errors made by the predictors on Data Set A depend strongly on the location in state space (see, for example, Smith, this volume). This is not surprising, since the local properties of the attractor (such as the rate of divergence of nearby trajectories) vary with the location. Furthermore, different regions are sampled unequally since the training set is finite.

The second assumption can be relaxed: in our own research, we have used maximum likelihood networks that successfully estimate the error bars as a function of the network input. These networks had two output units: the first one predicts the value, the second the error bar. This model of a single Gaussian (where mean and width are estimated depending on the input) is easily generalized to a mixture of Gaussians. It is also possible to explore more general models with the activation distributed over a set of output units, allowing one to predict the probability density.[24] All these models fall within the elegant and powerful probabilistic framework laid out for connectionist modeling by Rumelhart et al. (1993). Another approach to estimating the output errors is to use "noisy weights": rather than characterizing each weight only by its value, Hinton and van Camp (1993) also encode the precision of the weight. In addition to facilitating the application of the minimum description length principle, this formulation yields a probability distribution of the output for a given input.

To force competition entrants to address the issue of the reliability of their estimates, we required them to submit, for Data Set A, both the predicted values and their estimated accuracies. We set as the error measure the likelihood of the true (withheld) data, computed from the submitted predictions and confidence intervals under a Gaussian model. In the Appendix we motivate this measure and list its values for the entries received. Analyzing the competition entries, we were quite surprised by how little effort was put into estimating the error bars. In all cases, the techniques used to generate the submitted error estimates were much less sophisticated than the models used for the point-predictions.

Motivated by this lack of sophistication, following the close of the competition, Fraser and Dimitriadis (this volume) used a hidden Markov model (HMM; see, e.g., Rabiner & Juang, 1986) to predict the evolution of the entire probability density for the computer-generated Data Set D, with 100,000 points the longest of the data sets. Nonhidden Markov models fall in the same class as feedforward networks: their only notion of state is what is presented to them in the moment—there is no implicit memory or stack. They thus implement a regular (although probabilistic) grammar. Hidden Markov models introduce "hidden states" that are not directly observable but are built up from the past. In this sense, they resemble recurrent networks which also form a representation of the past in internal memory that can influence decisions beyond what is seen by the immediate inputs. Both HMMs and recurrent networks thus implement context-free grammars.

[23] Fraser (personal communication, 1993) points out that this argument can be traced back to Rissanen and Langdon (1981).

[24] On the one hand, a localist representation is suited for the prediction of symbols where we want to avoid imposing a metric. For example, in composing music, it is undesirable to inherit a metric obtained from the pitch value (see Dirst & Weigend, this volume). On the other hand, if neighborhood is a sensible distance measure, then higher accuracy can be obtained by smearing out the activation over several units (e.g., Saund, 1989).

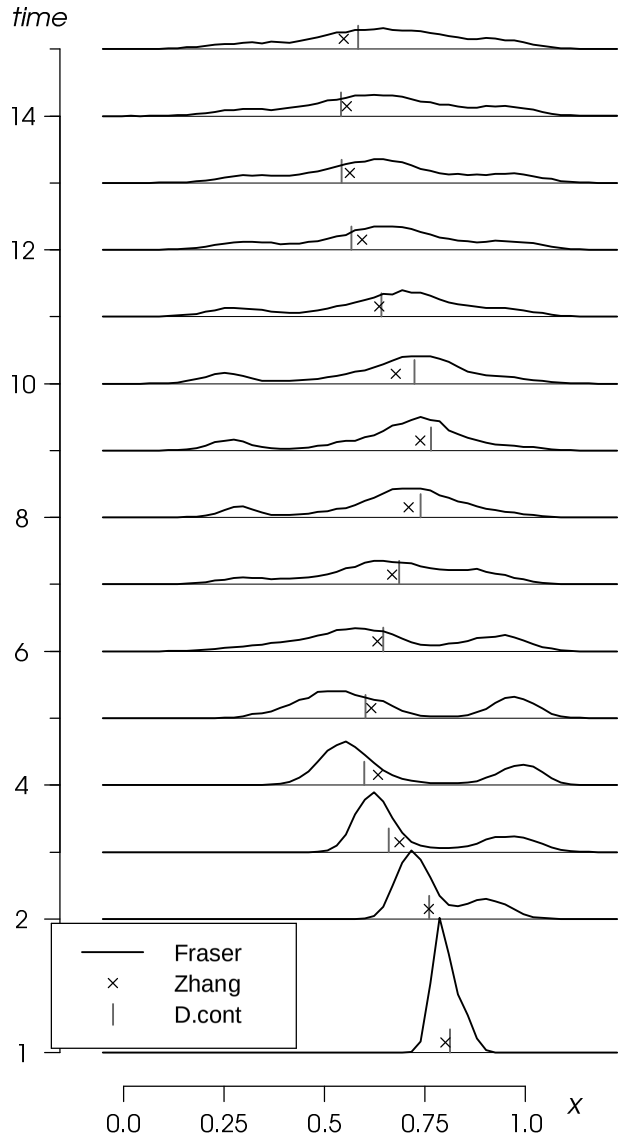

FIGURE 7 Continuations of Data Set D, the computer-generated data. The curves are the predictions for the probability density function from a hidden Markov model (Fraser & Dimitriadis, this volume). The ×'s indicate the point predictions from a neural network (Zhang & Hutchinson, this volume). The "true" continuation is indicated by vertical lines.

The states of a hidden Markov model can be discrete or continuous; Fraser and Dimitriadis use a mixture of both: they generate their predictions from a 20-state model using eighth-order linear autoregressive filters. The model parametrizes multivariate normal distributions.

Figure 7 shows the evolution of the probability density function according to Fraser and Dimitriadis (this volume). They first estimated the roughly 6,000 parameters of their model from the 100,000 points of Data Set D, and then generated several million continuations from their model. We plot the histograms of these continuations (normalized to have the same areas) along with the values obtained by continuing the generation of the data from the differential equations (as described in the Appendix). Unfortunately, the predicted probability density functions by Fraser and Dimitriadis are a lot wider than what the uncertainty due to the dynamics of the system and the stochasticity used in generating the series requires: on the time scale of Figure 7, an ensemble of continuations of Data Set D spreads only about 1%, a lot less than the uncertainty predicted by the mixed-state HMM.

In the competition, we received several sets of point-predictions for Data Set D. They are listed and plotted in the Appendix. In Figure 7 we have included the best of these, obtained by Zhang and Hutchinson (this volume). They trained 108 simple feedforward networks in a brute force approach: the final run alone—after all initial experimentation and architecture selection—used 100 hours on a Connection Machine CM-2 with 8,192 floating point processors. Each network had between 20 and 30 inputs, either one hidden layer with 100 units or two layers with 30 units each, and up to 5 outputs. The complex, unstructured architecture unfortunately does not allow a satisfying interpretation of the resulting networks.

Beyond physical systems: Financial data. We close this section with some remarks about the forecasts for the exchange rate time series (Data Set C). (No prediction tasks were specified for the biological, astronomical, and musical sets.) The market is quite large: in 1989 the daily volume of the currency markets was estimated to be U.S. $650 billion, and in 1993 the market exceeded U.S. $1 trillion ($= 10^{12}$) on busy days. Lequarré (this volume) reports that 97% of this is speculation—i.e., only 3% of the trades are "actual" transactions.

The simplest model of a market indicator is the efficient market hypothesis: "you can't do better than predicting that tomorrow's rate is the same as today's." Diebold and Nason (1990), for example, review attempts to forecast foreign exchange rates and conclude that none of the methods succeeded in beating the random walk hypothesis out-of-sample. However, all these academic findings about the unpredictability of the vicissitudes of financial markets are based on daily or weekly rates, the only data available until a few years ago. The great deal of activity in foreign exchange trading suggests that it must be possible to make money with better data.

Recently, high-frequency data have become available, and we were fortunate enough to be able to provide a set of such "tick-by-tick" data for the competition.

Data Set C consists of quotes on the time scale of one to two minutes for the exchange rate between the Swiss franc and the U.S. dollar. The market is based on bids and asks to buy and sell. (There is no central market for currency transactions.) Prices are given for a trade of U.S. $10 million, and an offer is good for five seconds(!). In addition to the quote, we included the day of the week and the time after the opening that day (to allow for modeling of intra- and inter-day effects).

In order to balance the desire for statistically significant results and the need to keep the competition prediction task manageable, we asked for forecasts for 10 episodes of 3,000 points each, taken from the period between August 7, 1990 to April 18, 1991.[25] This assessment provided a basic sanity test of the submitted predictors (some of them were worse than chance by a factor of 16), and afterwards the best two groups were invited to analyze a much larger sample. The quality of the predictions is expressed in terms of the following ratio of squared errors:

$$\frac{\sum_t \left(\text{observation}_t - \text{prediction}_t\right)^2}{\sum_t \left(\text{observation}_t - \text{observation}_{t-1}\right)^2} \,. \tag{24}$$

The denominator simply predicts the last observed value—which is the best that can be done for a random walk. A ratio above 1.0 thus corresponds to a prediction that is worse than chance; a ratio below 1.0 is an improvement over a random walk.[26]

In this second round of evaluation, predictions were made for all the points in the gaps between the training segments (still using the competition data for training). The out-of-sample performance on this extended test set, as reported by Mozer (this volume) and by Zhang and Hutchinson (this volume), is collected in Table 1. Please refer to their articles for more details and further evaluation.

TABLE 1 Performance on exchange rate predictions expressed as the squared error of the predictor divided by the squared error from predicting no change, as defined in Eq. (24). The numbers are as reported by Mozer (this volume, p. 261) for his recurrent networks, and Zhang and Hutchinson (this volume, p. 236) for their feedforward networks.

	1 minute ($N = 18,465$)	15 minutes ($N = 7,246$)	1 hour ($N = 3,334$)
Mozer	0.9976	0.9989	0.9965
Zhang & Hutchinson	1.090	1.103	1.098

These results—in particular the last row in Table 1 where all out-of-sample predictions are on average worse than chance—make clear that a naive application of the techniques that worked so well for Data Set A (the laser had less than 1% measurement noise added to deterministic dynamics) and to some degree for Data Set D fails for a data set so close to pure randomness as this financial data set. Future directions include (Weigend, 1991)

- using additional information from other sources (such as other currencies, interest rates, financial indicators, as well as information automatically extracted from incoming newswires, "topic spotting"),
- splitting the problem of predicting returns to the two separate tasks of predicting the squared change (volatility) and its direction,
- employing architectures that allow for stretching and compressing of the time series, as well as enhancing the input with features that collapse time in ways typically done by traders,
- implementing trading strategies (i.e., converting predictions to recommendations for actions), and subsequently improving them by backpropagating the actual loss or profit through a pay-off matrix (taking transaction costs into account).

Financial predictions can also serve as good vehicles to stimulate research in areas such as subset selection (finding relevant variables) and capacity control (avoiding overfitting). The full record of 329,112 quotes (bid and ask) from May 20, 1985 to April 12, 1991 is available as a benchmark data set via anonymous ftp.[27] We encourage the reader to experiment (and inform the authors of positive results).

[25] For each trial we asked for six forecasts: 1 minute after the last tick, 15 minutes after the last tick, 60 minutes after the last tick, the closing value of the day of the last tick, the opening value of the next trading day, and the closing value of the fifth trading day (usually one week) after the day of the last tick. One example evaluation is given in the following table. The numbers are the ratio of the sum of squared errors of the submitted predictions by Zhang and Hutchinson (this volume, p. 235) divided by the sum of the squared errors obtained by using the last observation as the prediction.

	1 minute	15 minutes	1 hour
training set ("in-sample")	0.889	0.891	0.885
test set ("out-of-sample")	0.697	1.04	0.988

This table shows the crucial difference between training set and test set performances, and suggests that the uncertainty in the numbers is large. Hence, we proposed the evaluation on larger data sets, as described in the main text.

[26] The random walk model used here for comparison is a weak null hypothesis. LeBaron (this volume) reports statistically significant autocorrelations on Data Set C (see Table 2 in his paper on p. 462 of this volume). To the degree that the in-sample autocorrelations generalize out-of-sample, this justifies a low-order AR model as a stronger null hypothesis. However, to avoid additional assumptions in our comparison here (such as the order of the model), we decided simply to compare to a random walk.

[27] It corresponds to 11.5 MB. Like the data sets used in the competition, it is available via anonymous ftp to `ftp.santafe.edu`.

6. CHARACTERIZATION

Simple systems can produce time series that appear to be complicated; complex systems can produce time series that are complicated. These two extremes have different goals and require different techniques; what constitutes a successful forecast depends on where the underlying system falls on this continuum. In this section we will look at characterization methods that can be used to extract some of the essential properties that lie behind an observed time series, both as an end in itself and as a guide to further analysis and modeling.

Characterizing time series through their frequency content goes back to Schuster's "periodogram" (1898). For a simple linear system the traditional spectral analysis is very useful (peaks = modes = degrees of freedom), but different nonlinear systems can have similar featureless broadband power spectra. Therefore, a broadly useful characterization of a nonlinear system cannot be based on its frequency content.

We will describe two approaches that parallel the earlier discussion of forecasting: (1) an explicit analysis of the structure in an embedding space (in Section 6.2 we will introduce the information-theoretic measure of redundancy as an example of understanding by "opening up the box"), and (2) an implicit approach based on analyzing properties of an emulation of the system arrived at through learning (in Section 6.3.1 we will show how local linear methods can be used for characterization, and in Section 6.3.2 we will discuss how neural networks can be used to estimate dimensions, the amount of nonlinearity, and Lyapunov coefficients). Before turning to these more sophisticated analyses, we discuss some simple tests.

6.1 SIMPLE TESTS

This section begins with suggestions for exploring data when no additional information is available. We then turn to time series whose frequency spectra follow a power law where low-frequency structure can lead to artifacts. We then show how surrogate data can be generated with identical linear but different nonlinear structure. Finally, we suggest some ways of analyzing the residual errors.

EXPLORATORY DATA ANALYSIS. The importance of exploring a given data set with a broad range of methods cannot be overemphasized. Besides the many traditional techniques (Tukey, 1977), modern methods of interactive exploration range from interactive graphics with linked plots and virtual movements in a visual space[28] to examination in an auditory space (data sonification). The latter method uses the temporal abilities of our auditory systems—after all, analyzing temporal sequences as static plots has not been a prime goal in human evolution.

[28] A good example is the visualization package `xgobi`. To obtain information about it, send the one line message `send index` to `statlib@lib.stat.cmu.edu`.

LINEAR CORRELATIONS. We have seen in Section 3.2 that nonlinear structure can be missed by linear analysis. But that is not the only problem: linear structure that is present in the data can confuse nonlinear analysis. Important (and notorious) examples are processes with power spectra proportional to $|\omega|^{-\alpha}$, which arise routinely in fluctuating transport processes. At the extreme, white noise is defined by a flat spectrum; i.e., its spectral coefficient is $\alpha = 0$. When white noise is integrated (summed up over time), the result is a random walk process. After squaring the amplitude (in order to arrive at the power spectrum), a random walk yields yields a spectral exponent $\alpha = 2$. The intermediate value of $\alpha = 1$ is seen in everything from electrical resistors to traffic on a highway to music (Dutta & Horn, 1981). This intermediate spectral coefficient of $\alpha = 1$ implies that all time-scales (over which the $1/\omega$ behavior holds) are equally important.

Consider the cloud of points obtained by plotting x_t against $x_{t-\tau}$ for such a series with a lag time τ. (This plot was introduced in Section 3.2 in the context of the logistic map whose phase portrait was a parabola.) In general, a distribution can be described by its moments. The first moments (means) give the coordinates of the center of the point cloud; the second moments (covariance matrix) contain the information about how elongated the point cloud is, and how it is rotated with respect to the axes.[29] We are interested here in this elongation of the point cloud; it can be described in terms of the eigenvalues of its correlation matrix. The larger eigenvalue, λ_+, characterizes the extension in the direction of the larger principal axis along the diagonal, and the smaller eigenvalue, λ_-, measures the extension transverse to it. The ratio of these two eigenvalues can be expressed in terms of the autocorrelation function ρ at the lag τ as

$$\frac{\lambda_-}{\lambda_+} = \frac{1-\rho(\tau)}{1+\rho(\tau)} \,. \tag{25}$$

For a power-law spectrum, the autocorrelation function can be evaluated analytically in terms of the spectral exponent α and the exponential integral Ei (Gershenfeld, 1992). For example, for a measurement bandwidth of 10^{-3} to 10^{3} Hz and a lag time τ of 1 sec, this ratio is 0.51 for $\alpha = 1$, 0.005 for $\alpha = 2$, and 0.0001 for $\alpha = 3$. As the spectrum drops off more quickly, i.e., as the spectral exponent gets larger, the autocorrelation function decays more slowly. (In the extreme of a delta function in frequency space, the signal is constant in time.) Large spectral exponents thus imply that the shape of the point cloud (an estimate of the probability distribution in the embedding space) will be increasingly long and skinny, regardless of the detailed dynamics and of the value of τ. If the width becomes small compared to

[29] We here consider only first- and second-order moments. They completely characterize a Gaussian distribution, and it is easy to relate them to the conventional (linear) correlation coefficient; see Duda and Hart (1973). A nonlinear relationship between x_t and $x_{t-\tau}$ is missed by an analysis in terms of the first- and second-order moments. This is why a restriction to up to second-order terms is sometimes called linear.

the available experimental resolution, the system will erroneously appear to be one-dimensional. There is a simple test for this artifact: whether or not the quantities of interest change as the lag time τ is varied. This effect is discussed in more detail by Theiler (1991).

SURROGATE DATA. Since the autocorrelation function of a signal is equal to the inverse Fourier transform of the power spectrum, any transformation of the signal that does not change the power spectrum will not change the autocorrelation function. It is therefore possible to take the Fourier transform of a time series, randomize the phases (symmetrically, so that the inverse transform remains real), and then take the inverse transform to produce a series that by construction has the same autocorrelation function but will have removed any nonlinear ordering of the points. This creation of sets of surrogate data provides an important test for whether an algorithm is detecting nonlinear structure or is fooled by linear properties (such as we saw for low frequency signals): if the result is the same for the surrogate data, then the result cannot have anything to do with deterministic rules that depend on the specific sequence of the points. This technique has become popular in recent years; Fraser (1989c), for example, compares characteristics of time series from the Lorenz equation with surrogate versions. Kaplan (this volume) and Theiler et al. (this volume) apply the method of surrogate data to Data Sets A, B, D, and E; the idea is also the basis of Paluš's comparison (this volume) between "linear redundancy" and "redundancy" (we will introduce the concept of redundancy in the next section).

SANITY CHECKS AND SMOKE ALARMS. Once a predictor has been fitted, a number of sanity checks should be applied to the resulting predictions and their residual errors. It can be useful to look at a distribution of the errors sorted by their size (see, e.g., Smith, this volume, Figures 5 and 8 on p. 331 and 336). Such plots distinguish between forecasts that have the same mean squared error but very different distributions (uniformly medium-sized errors versus very small errors along with a few large outliers). Other basic tests plot the prediction errors against the true (or against the predicted) value. This distribution should be flat if the errors are Gaussian distributed, and proportional to the mean for a Poissonian error distribution. The time ordering of the errors can also contain information: a good model should turn the time series into structureless noise for the residual errors; any remaining structure indicates that the predictor missed some features.

Failure to use common sense was readily apparent in many of the entries in the competition. This ranged from forecasts for Data Set A that included large negative values (recall that the training set was strictly positive) to elaborate "proofs" that Data Set D was very low dimensional (following a noise reduction step that had the effect of removing the high-dimensional dynamics). Distinguishing fact, fiction and fallacies in forecasts is often hard: carrying out simple tests is crucial particularly

in the light of readily available sophisticated analysis and prediction algorithms that can swiftly and silently produce nonsense.

6.2 DIRECT CHARACTERIZATION VIA STATE SPACE

It is always possible to define a time-delayed vector from a time series, but this certainly does not mean that it is always possible to identify meaningful structure in the embedded data. Because the mapping between a delay vector and the system's underlying state is not known, the precise value of an embedded data point is not significant. However, because an embedding is diffeomorphic (smooth and invertible), a number of important properties of the system will be preserved by the mapping. These include local features such as the number of degrees of freedom, and global topological features such as the linking of trajectories (Melvin & Tufillaro, 1991). The literature on characterizing embedded data in terms of such invariants is vast, motivated by the promise of obtaining deep insight into observations, but plagued by the problem that plausible algorithms will always produce a result—whether or not the result is significant. General reviews of this area may be found in Ruelle and Eckmann (1985), Gershenfeld (1989), and Theiler (1990).[30]

Just as state-space forecasting leaps over the systematic increase in complexity from linear models to bilinear models, etc., these characterization ideas bypass the traditional progression from ordinary spectra to higher order spectra.[31] They are predated by similar efforts to analyze signals in terms of dimensionality (Trunk, 1968) and near-neighbor scaling (Pettis et al., 1979), but they could not succeed until the relationship between observed and unobserved degrees of freedom was made explicit by time-delay embedding. This was done to estimate degrees of freedom by Russel et al. (1980), and implemented efficiently by Grassberger and Procaccia (1983a). Brock, Dechert, and Scheinkman later developed this algorithm into a statistical test, the BDS test (Brock et al., 1988), with respect to the null hypothesis of an iid sequence (which was further refined by Green & Savit, 1991).

We summarize here an information-based approach due to Fraser (1989b) that was successfully used in the competition by Paluš (this volume) (participating in the competition over the network from Czechoslovakia). Although the connection between information theory and ergodic theory has long been appreciated (see, e.g., Petersen, 1989), Shaw (1981) helped point out the connection between dissipative dynamics and information theory, and Fraser and Swinney (1986) first used information-theoretic measures to find optimal embedding lags. This example of

[30] The term "embedding" is used in the literature in two senses. In its wider sense, the term denotes any lag-space representation, whether there is a unique surface or not. In its narrower (mathematical) sense used here, the term applies if and only if the resulting surface is unique, i.e., if a diffeomorphism exists between the solution manifold in configuration space and the manifold in lag space.

[31] Chaotic processes are analyzed in terms of bispectra by Subba Rao (1992).

the physical meaning of information (Landauer, 1991) can be viewed as an application of information theory back to its roots in dynamics: Shannon (1948) built his theory of information on the analysis of the single-molecule Maxwell Demon by Szilard in 1929, which in turn was motivated by Maxwell and Boltzmann's effort to understand the microscopic dynamics of the origin of thermodynamic irreversibility (circa 1870).

Assume that a time series $x(t)$ has been digitized to integer values lying between 1 and N. If a total of n_T points have been observed, and a particular value of x is recorded n_x times, then the probability of seeing this value is estimated to be $p_1(x) = n_x/n_T$.[32] (The subscript of the probability indicates that we are at present considering one-dimensional distributions (histograms). It will soon be generalized to d-dimensional distributions.) In terms of this probability, the **entropy** of this distribution is given by

$$H_1(N) = -\sum_{x=1}^{N} p_1(x) \log_2 p_1(x) \,. \tag{26}$$

This is the average number of bits required to describe an isolated observation, and can range from 0 (if there is only one possible value for x) to $\log_2 N$ (if all values of x are equally likely and hence the full resolution of x is required).

In the limit $N \to 1$, there is only one possible value and so the probability of seeing it is unity, thus $H_1(1) = 0$. As N is increased, the entropy grows as $\log N$ if all values are equally probable; it will reach an asymptotic value of $\log M$ independent of N if there are M equally probable states in the time series; and if the probability distribution is more complicated, it can grow as $D_1 \log N$ where D_1 is a constant ≤ 1 (the meaning of D_1 will be explained shortly). Therefore, the dependence of H_1 on N provides information about the resolution of the observable.

The probability of seeing a specific lag vector $\mathbf{x}_t = (x_t, x_{t-\tau}, \ldots, x_{t-(d-1)\tau})$ (see Eq. (14)) in d-dimensional lag space is similarly estimated by counting the relative population of the corresponding cell in the d-dimensional array: $p_d(\mathbf{x}) = n_{\mathbf{x}}/n_T$. The probability of seeing a particular *sequence* of D embedded vectors $(\mathbf{x}_t, \ldots, \mathbf{x}_{t-(D-1)\tau})$ is just $p_{d+D}(x_t, \ldots, x_{t-(d+D-1)\tau})$ because each successive vector is equal to the preceding one with the coordinates shifted over one place and a new observation added at the end. This means that the joint probability of d delayed observations, p_d, is equivalent to the probability of seeing a single point in the d-dimensional embedding space (or the probability of seeing a sequence of $1 + d - n$ points in a smaller n-dimensional space). In terms of p_d, the **joint entropy** or **block entropy** is

$$H_d(\tau, N) = -\sum_{x_t=1}^{N} \cdots \sum_{x_{t-(d-1)\tau}=1}^{N} p_d(x_t, x_{t-\tau}, \ldots, x_{t-(d-1)\tau}) \log_2 p_d(x_t, x_{t-\tau}, \ldots, x_{t-(d-1)\tau}) \,. \tag{27}$$

This is the average number of bits needed to describe a sequence. (The range of the sum might seem strange at first sight, but keep in mind that we are assuming that x_t is quantized to integers between 1 and N.) In the limit of small lags, we obtain

$$\begin{aligned} \lim_{\tau \to 0} p_d(x_t, x_{t-\tau}, \ldots, x_{t-(d-1)\tau}) = p_1(x) \\ \Rightarrow H_d(0, N) = H_1(N) \,. \end{aligned} \tag{28}$$

In the opposite limit, if successive measurements become uncorrelated at large times, the probability distribution will factor:

$$\begin{aligned} \lim_{\tau \to \infty} p_d(x_t, x_{t-\tau}, \ldots, x_{t-(d-1)\tau}) = p_1(x_t) p_1(x_{t-\tau}), \ldots, p_1(x_{t-(d-1)\tau}) \\ \Rightarrow \lim_{\tau \to \infty} H_d(\tau, N) = d H_1(N) \,. \end{aligned} \tag{29}$$

We will return to the τ dependence later; for now assume that the delay time τ is small but nonzero.

We have already seen that $\lim_{N \to 1} H_d(\tau, N) = 0$. The limit of large N is best understood in the context of the **generalized dimensions** D_q (Hentschel & Procaccia, 1983). These are defined by the scaling of the moments of the d-dimensional probability distribution $p_d(\mathbf{x})$ as the number of bins N tends to infinity (i.e., the bin sizes become very small, corresponding to an increasing resolution):

$$D_q = \lim_{N \to \infty} \frac{1}{q-1} \frac{\log_2 \sum_{\mathbf{x}_i} p_d(\mathbf{x}_i)^q}{-\log_2 N} \,. \tag{30}$$

For simple geometrical objects such as lines or surfaces, the D_q's are all equal to the integer topological dimension (1 for a line, 2 for a surface,...). For fractal distributions they need not be an integer, and the q dependence is related to how singular the distribution is. The values of the D_q's are typically similar, and there are strong bounds on how much they can differ (Beck, 1990). D_2 measures the scaling of pairs of points (it is the Grassberger-Procaccia correlation dimension (Grassberger & Procaccia, 1983a); see also Kantz, this volume, and Pineda & Sommerer, this volume), and D_1 provides the connection with entropy:

$$\begin{aligned} \lim_{q \to 1} D_q &= \lim_{N \to \infty} \frac{\sum_{\mathbf{x}_i} p_d(\mathbf{x}_i) \log_2 p_d(\mathbf{x}_i)}{-\log_2 N} \\ &= \lim_{N \to \infty} \frac{H_d(\tau, N)}{\log_2 N} \,. \end{aligned} \tag{31}$$

[32] Note that there can be corrections to such estimates if one is interested in the expectation value of functions of the probability (Grassberger, 1988).

As N is increased, the prefactor to the logarithmic growth of the entropy is the generalized dimension D_1 of the probability distribution. If the system is deterministic, so that its solutions lies on a low-dimensional attractor, the measured dimension D_1 will equal the dimension of the attractor if the number of time delays used is large enough. If the number of lags is too small, or if the successive observations in the time series are uncorrelated, then the measured dimension will equal the number of lags. The dimension of an attractor measures the number of local directions available to the system and so it (or the smallest integer above it if the dimension is fractal) provides an estimate of the number of degrees of freedom needed to describe a state of the system. If the underlying system has n degrees of freedom, the minimum embedding dimension to recover these dynamics can be anywhere between n and $2n$, depending on the geometry.

The d-dependence of the entropy can be understood in terms of the concept of **mutual information**. The mutual information between two samples is the difference between their joint entropy and the sum of their scalar entropies:

$$\begin{aligned} I_2(\tau, N) = &- \sum_{x_t=1}^{N} p_1(x_t) \log_2 p_1(x_t) - \sum_{x_{t-\tau}=1}^{N} p_1(x_{t-\tau}) \log_2 p_1(x_{t-\tau}) \\ &+ \sum_{x_t=1}^{N} \sum_{x_{t-\tau}=1}^{N} p_2(x_t, x_{t-\tau}) \log_2 p_2(x_t, x_{t-\tau}) \\ &= 2H_1(\tau, N) - H_2(\tau, N)\,. \end{aligned} \tag{32}$$

If the samples are statistically independent (this means by definition that the probability distribution factors, i.e., $p_2(x_t, x_{t-\tau}) \equiv p_1(x_t)p_1(x_{t-\tau})$), then the mutual information will vanish: no knowledge can be gained for the second sample by knowing the first. On the other hand, if the first sample uniquely determines the second sample ($H_2 = H_1$), the mutual information will equal the scalar entropy $I_2 = H_1$. In between these two cases, the mutual information measures in bits the degree to which knowledge of one variable specifies the other.

The mutual information can be generalized to higher dimensions either by the **joint mutual information**

$$I_d(\tau, N) = dH_1(\tau, N) - H_d(\tau, N) \tag{33}$$

or by the **incremental mutual information** or **redundancy** of one sample

$$R_d(\tau, N) = H_1(\tau, N) + H_{d-1}(\tau, N) - H_d(\tau, N)\,. \tag{34}$$

The redundancy measures the average number of bits about an observation that can be determined by knowing $d-1$ preceding observations. Joint mutual information and redundancy are related by $R_d = I_d - I_{d-1}$.

For systems governed by differential equations or maps, given enough points and enough resolution, the past must uniquely determine the future[33] (to a certain horizon, depending on the finite resolution).

If d is much less than the minimum embedding dimension, then the $d-1$ previous observations do not determine the next one, and so the value of the redundancy will approach zero:

$$\begin{aligned} p_d(x_t, x_{t-\tau}, \ldots, x_{t-(d-1)\tau}) &= p_1(x_t) p_{d-1}(x_{t-\tau}, \ldots, x_{t-(d-1)\tau}) \\ &\Rightarrow H_d = H_1 + H_{d-1} \Rightarrow R_d = 0\,. \end{aligned} \tag{35}$$

On the other hand, if d is much larger than the required embedding dimension, then the new observation will be entirely redundant:[34]

$$\begin{aligned} p_d(x_t, x_{t-\tau}, \ldots, x_{t-(d-1)\tau}) &= p_{d-1}(x_{t-\tau}, \ldots, x_{t-(d-1)\tau}) \\ &\Rightarrow H_d = H_{d-1} \Rightarrow R_d = H_1\,. \end{aligned} \tag{36}$$

The minimum value of d for which the redundancy converges (if there is one) is equal to the minimum embedding dimension at that resolution and delay time, i.e., the size of the smallest Euclidean space that can contain the dynamics without trajectories crossing. Before giving the redundancy for Data Set A of the competition (in Figure 8), we provide relations to other quantities, such as the Lyapunov exponents and the prediction horizon.

The **source entropy** or **Kolmogorov-Sinai entropy**, $h(\tau, N)$, is defined to be the asymptotic rate of increase of the information with each additional measurement given unlimited resolution:

$$h(\tau, N) = \lim_{N\to\infty} \lim_{d\to\infty} H_d(\tau, N) - H_{d-1}(\tau, N)\,. \tag{37}$$

The limit of infinite resolution is usually not needed in practice: the source entropy reaches its maximum asymptotic value once the resolution is sufficiently fine to produce a generating partition (Petersen, 1989, p. 243). The source entropy is

[33] Note that the converse remains true for differential equations but need not be true for maps: for example, neither of the two maps introduced in Section 3.2 has a unique preimage in time—they cannot be inverted. Given that most computer simulations approximate the continuous dynamics of differential equations by discrete dynamics, this is a potentially rich source for artifacts. Lorenz (1989) shows how discretization in time can create dynamical features that are impossible in the continuous-time system; see also Grebogi et al. (1990). Rico-Martinez, Kevrekidis, and Adomaitis (1993) discuss noninvertibility in the context of neural networks.

[34] To be precise, $H_d = H_{d-1}$ is only valid for (1) the limit of short times τ, (2) discrete measurements, and (3) the noise-free case.

important because the *Pesin identity* relates it to the sum of the positive Lyapunov exponents (Ruelle & Eckmann, 1985):

$$h(\tau) = \tau h(1) = \tau \sum_i \lambda_i^+ . \tag{38}$$

The **Lyapunov exponents** λ_i are the eigenvalues of the local linearization of the dynamics (i.e., a local linear model), measuring the average rate of divergence of the principal axes of an ensemble of nearby trajectories. They can be found from the Jacobian (if it is known) or by following trajectories (Brown et al., 1991). Diverging trajectories reveal information about the system which is initially hidden by the measurement quantization. The amount of this information is proportional to the expansion rate of the volume, which is given by the sum of the positive exponents.

If the sequence of a time series generated by differential equations is reversed, the positive exponents will become negative ones and vice versa, and so the sum of negative exponents can be measured on a reversed series (Parlitz, 1992). If the time series was produced by a discrete map rather than a continuous flow (from differential equations), then the governing equations need not be reversible; for such a system the time-reversed series will no longer be predictable. Therefore, examining a series backward as well as forward is useful for determining whether a dynamical system is invertible and, if it is, the rate at which volumes contract. Note that reversing a time series is very different from actually running the dynamics backwards; there may not be a natural measure for the backwards dynamics and, if there is one, it will usually not be the same as that of the forwards dynamics. Since Lyapunov exponents are defined by time averages (and hence with respect to the natural measure), they will also change.

If the embedding dimension d is large enough, the redundancy is just the difference between the scalar entropy and an estimate of the source entropy:

$$R_d(\tau, N) \approx H_1(\tau, N) - h(\tau, N) . \tag{39}$$

In the limit of small lags,

$$H_{d-1}(0, N) = H_d(0, N) \Rightarrow R_d(0, N) = H_1(N) , \tag{40}$$

and for long lags

$$\lim_{\tau \to \infty} H_d(\tau, N) = d H_1(\tau, N) \Rightarrow R_d(\infty, N) = 0 . \tag{41}$$

The value of τ where the redundancy vanishes provides an estimate of the limit of predictability of the system at that resolution (prediction horizon) and will be very short if d is less than the minimum embedding dimension. Once d is larger than the embedding dimension (if there is one), then the redundancy will decay much more slowly, and the slope for small τ will be the source entropy:

$$R_d(\tau, N) = H_1(N) - \tau h(1) . \tag{42}$$

We have seen that it is possible from the block entropy (Eq. (27)) and the redundancy (Eq. (34)) to estimate the resolution, minimum embedding dimension, information dimension D_1, source entropy, and the prediction horizon of the data, and hence learn about the number of degrees of freedom underlying the time series and the rate at which it loses memory of its initial conditions. It is also possible to test for nonlinearity by comparing the redundancy with a linear analog defined in terms of the correlation matrix (Paluš, this volume). The redundancy can be efficiently computed with an $O(N)$ algorithm by sorting the measured values on a simple fixed-resolution binary tree (Gershenfeld, 1993b). This tree sort is related to the frequently rediscovered fact that box-counting algorithms (such as are needed for estimating dimensions and entropies) can be implemented in high-dimensional space with an $O(N \log N)$ algorithm requiring no auxiliary storage by sorting the appended indices of lagged vectors (Pineda & Sommerer, this volume). Equal-probability data structures can be used (at the expense of computational complexity) to generate more reliable unbiased entropy estimates (Fraser & Swinney, 1986).

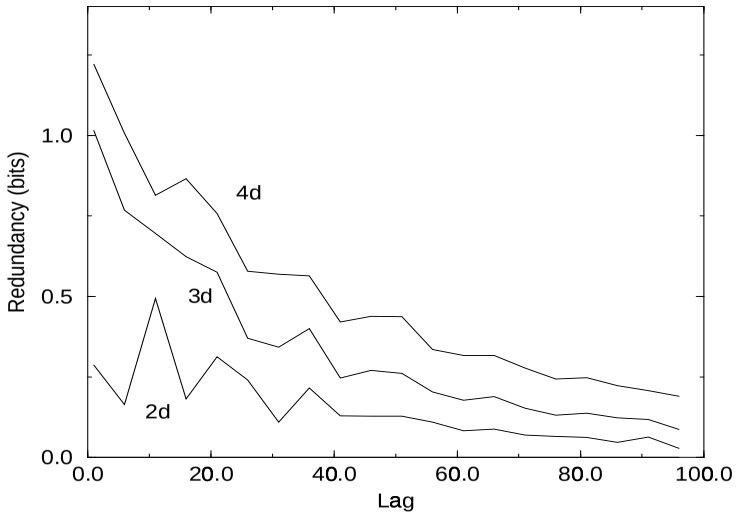

FIGURE 8 The redundancy (incremental mutual information) of Data Set A as a function of the number of time steps. The figure indicates that three past values are sufficient to retrieve most of the predictable structure and that the system has lost the memory of its initial conditions after roughly 100 steps.

Figure 8 shows the results of our redundancy calculation for Data Set A. This figure was computed using 10,000 data points sorted on the three most significant bits. Note that three lags are sufficient to retrieve most of the predictable structure. This is in agreement with the exploratory stereo pairs (Figure 5), where the three dimensions plotted appear to be sufficient for an embedding. Furthermore, we can read off from Figure 8 that the system has lost memory of its initial condition after about 100 steps.

There are many other approaches to characterizing embedded data (and choosing embedding parameters): Liebert and Schuster (1989) comment on a good choice for the delay time by relating the first minimum of mutual information plotted as a function of the lag time (suggested by Fraser & Swinney, 1986) to the generalized correlation integral. Aleksić (1991) plots the distances between images of close points as a function of the number of lags: at the minimum embedding dimension the distance suddenly becomes small. Savit and Green (1991), and Pi and Peterson (1993) exploit the notion of continuity of a function (of the conditional probabilities in the lag space as the number of lags increases). Kennel, Brown, and Abarbanel (1992) look for "false neighbors" (once the space is large enough for the attractor to unfold, they disappear). Further methods are presented by Kaplan, and by Pineda and Sommerer in this volume. Gershenfeld (1992) discusses how high-dimensional probability distributions limit the most complicated deterministic system that can be distinguished from a stochastic one. (The expected fraction of "typical" points in the interior of a distribution is increased; this is true because the ratio of the volume of a thin shell near the surface to the overall volume tends to 1 with increasing dimension.) Work needs to be done to understand the relations, strengths, and weaknesses of these algorithms. However, regardless of the algorithm employed, it is crucial to understand the nature of the errors in the results (both statistical uncertainty and possible artifacts), and to remember that there is no "right" answer for the time delay τ and the number of lags d—the choices will always depend on the goal.

The entries for the analysis part of the competition showed good agreement in the submitted values of the correlation dimension for Data Set A (2.02, 2.05, 2.06, 2.07, 2.2), but the estimates of the the positive Lyapunov exponent (either directly or from the source entropy) were more scattered (.024, .037, .07, .087, .089 bits/step). There were fewer estimates of these quantities for the other data sets (because they were more complicated) and they were more scattered; estimates of the degrees of freedom of Data Set D ranged from 4 to 8.

Insight into embedding can be used for more than characterization; it can also be used to distinguish between measurement noise and intrinsic dynamics. If the system is known, this can be done with a Wiener filter, a two-sided linear filter that estimates the most likely current value (rather than a future value). This method, however, requires advance knowledge of the power spectra of both the desired signal and of the noise, and the recovery will be imperfect if these spectra overlap (Priestley, 1981, p. 775; Press, 1992, p. 574). State-space reconstruction through time-average expectations (Eq. (16)) provides a method for signal separation that requires information only about the noise. If the true observable $x(t)$ is corrupted by additive measurement noise $n(t)$ that is uncorrelated with the system, then the expectation will factor:

$$\langle e^{i\mathbf{k}\cdot(\mathbf{x}(t)+\mathbf{n}(t))}\rangle = \langle e^{i\mathbf{k}\cdot\mathbf{x}(t)}\rangle\langle e^{i\mathbf{k}\cdot\mathbf{n}(t)}\rangle\,; \tag{43}$$

$\mathbf{k}$ is the wave vector indexing the Fourier transform of the state-space probability density. The noise produces a $\mathbf{k}$-dependent correction in the embedding space; if the noise is uncorrelated with itself on the time scale of the lag time τ (as for additive white noise), then this correction can be estimated and removed solely from knowledge of the probability distribution for the noise (Gershenfeld, 1993a). This algorithm requires sufficient data to accurately estimate the expectation values; Marteau and Abarbanel (1991), Sauer (1992), and Kantz (this volume) describe more data-efficient alternatives based on distinguishing between the low-dimensional system's trajectory and the high-dimensional behavior associated with the noise. Much less is known about the much harder problem of separating noise that enters into the dynamics (Guckenheimer, 1982).

6.3 INDIRECT CHARACTERIZATION: UNDERSTANDING THROUGH LEARNING

In Section 3.1 we showed that a linear time system is fully characterized by its Fourier spectrum (or equivalently by its ARMA coefficients or its autocorrelation function). We then showed how we have to go beyond that in the case of nonlinear systems and focused in Section 6.2 on the properties of the observed points in embedding space. As with forecasting, we move from the direct approach to the case where the attempt to understand the system directly does not succeed: we now show examples of how time series can be characterized through forecasting. The price for this appealing generality will be less insight into the meaning of the results. Both classes of algorithms that were successful in the competition will be put to work for characterization; in Section 6.3.1 we use local linear models to obtain DVS plots ("deterministic vs. stochastic") and, in Section 6.3.2, we explain how properties of the system that are not directly accessible can be extracted from connectionist networks that were trained to emulate the system.

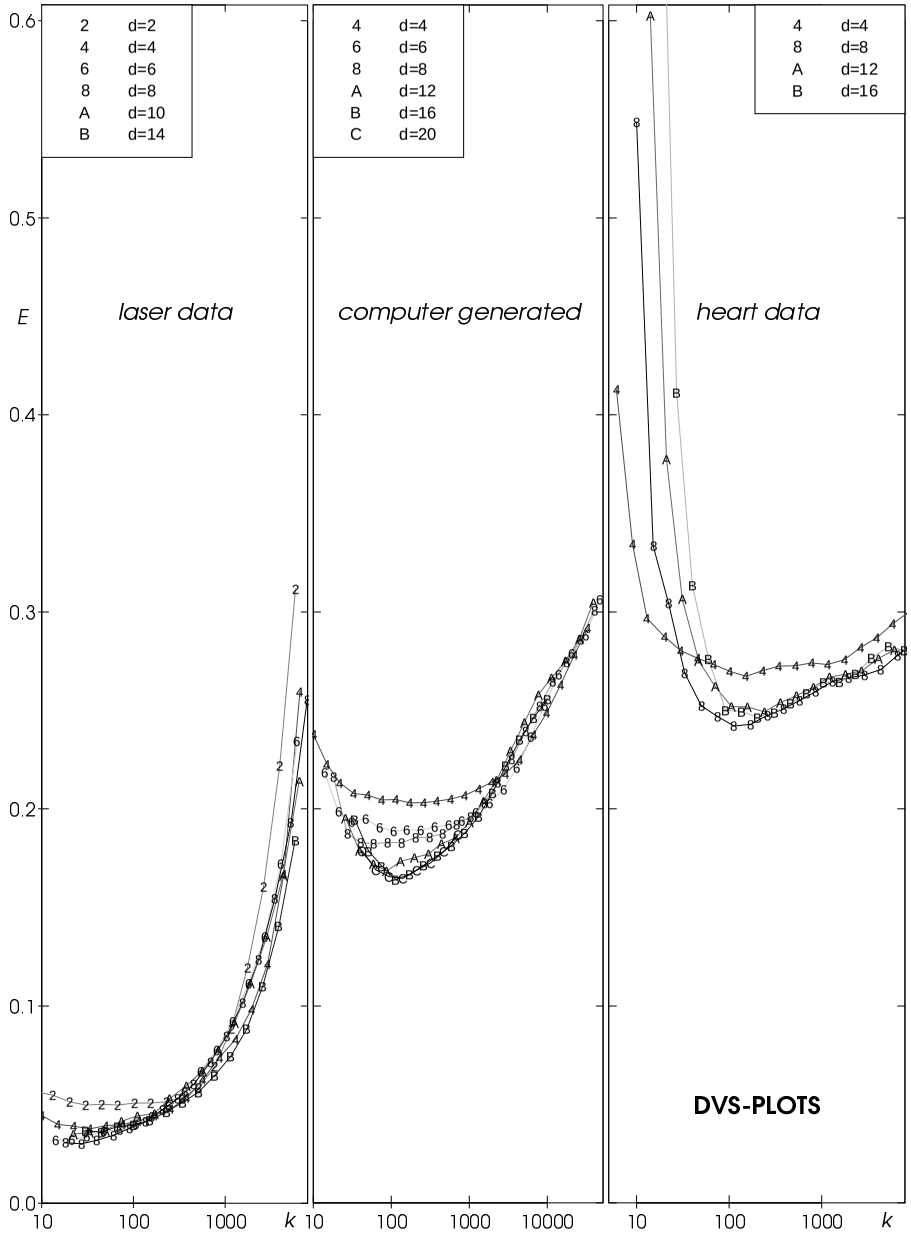

FIGURE 9 Deterministic vs. stochastic plots: The normalized out-of-sample error E is shown as function of the number of neighbors k used to construct a local linear model of order d.

6.3.1 CHARACTERIZATION VIA LOCAL LINEAR MODELS: DVS PLOTS. Forecasting models often possess some "knobs" that can be tuned. The dependence of the prediction error on the settings of these knobs can sometimes reveal—indirectly—some information about properties of the system. In local linear modeling, examples of such knobs are the number of delay values d and the number of neighbors k used to construct the local linear model. Casdagli (1991) introduces the term "deterministic vs. stochastic modeling" (DVS) for the turning of these knobs, and Casdagli and Weigend (this volume) apply the idea to the competition data.

In Figure 9 we show the out-of-sample performance for a local linear model on three of the Santa Fe data sets (laser data `A.con`, computer-generated data `D1.dat`, and the heart data `B2.dat`) as a function of the number of neighbors k used for the linear fit. The left side of each plot corresponds to a simple look-up of the neighbor closest in lag space; the right corresponds to a global linear model that fits a hyperplane through all points. In all three panels the scale of the y-axis (absolute errors) is the same. (Before applying the algorithm, all series were normalized to unit variance.)

The first observation is the overall size of the out-of-sample errors:[35] The laser data are much more predictable than the computer-generated data, which in turn are more predictable than the heart data. The second observation concerns the shape of the three curves. The location of the minimum shifts from the left extreme for the laser (next-neighbor look-up), through a clear minimum for the computer-generated data (of about one hundred neighbors), to a rather flat behavior beyond a sharp drop for the heart data.

So far, we have framed this discussion along the axis (local linear)↔(global linear). We now stretch the interpretation of the fit in order to infer properties of the generating system. Casdagli (1991) uses the term "deterministic" for the left extreme of local linear models, and "stochastic" for the right extreme of global linear models. He motivates this description with computer-generated examples as well as with experimental data from cellular flames, EEG, fully developed turbulence, and the sunspot series. Deterministic nonlinear systems are often successfully modeled by local linear models: the error is small for small neighborhoods, but increases as more and more neighbors are included in the fit. When the neighborhood size is too large, the hyperplane does not accurately approximate the (nonlinear) manifold of the data, and the out-of-sample error increases. This is indeed the case for the laser. The local linear forecasts are 5 to 10 times more accurate than global linear

[35] We give average absolute errors since they are more robust than squared errors, but the qualitative behavior is the same for squared errors. The numerical values are obtained by dividing the sum of these "linear" errors by the size of the test example (500 points in all three series). Furthermore, the delay between each past sample ("lag time") is chosen to be one time step in all three series, but the prediction time ("lead time") is chosen to be one time step for the laser, two time steps for the computer-generated data, and four time steps for the heart rate predictions to reflect the different sampling rates with respect to the timescale of the dynamics. The predictions for the heart data were obtained from a bivariate model, using both the heart rate and the chest volume as input. The details are given in the article by Casdagli and Weigend (this volume).

ones, suggesting the interpretation of the laser as a nonlinear deterministic chaotic system. On the other hand, if a system is indeed linear and stochastic, then using smaller neighborhoods makes the out-of-sample predictions worse, due to overfitting of the noise. This is apparent for the heart data, which clearly shows overfitting for small neighborhood sizes and therefore rules out simple deterministic chaos. The DVS plot alone is not powerful enough to decide whether nonlinearities are present in the system or not.[36] Finally, the middle example of computer-generated data falls between these two cases. Nonlinearity, but not low-dimensional chaos, is suggested here since the short-term forecasts at the minimum are between 50% to 100% more accurate than global linear models.

Apart from the number of neighbors, the other knob to turn is the order of the linear model, i.e., the number of time delays d. The order of the AR model that makes successful predictions provides an upper bound on the minimum embedding dimension. For the laser, $d = 2$ is clearly worse than $d = 4$, and the lowest out-of-sample errors are reached for $d = 6$. This indeed is an upper estimate for $d = 3$ (compare to Figures 5 and 8). For the computer-generated data, the quality of the predictions continually increases from $d = 4$ to $d = 12$ and saturates at $d = 16$. For the heart data—since the DVS plots give no indications of low-dimensional chaos—it does not make sense to give an embedding dimension into which the geometry can be disambiguated.

We close this section with a statistical perspective on DVS plots: although their interpretation is necessarily somewhat qualitative, they nicely reflect the trade-off between bias and variance. A weak local linear model has a low bias (it is very flexible), but the parameters have a high variance (since there are only a few data points for estimating each parameter). A strong global linear model has a large model bias, but the parameters can be estimated with a small variance since many data points are available to determine the fit (see, e.g., Geman, Bienenstock, & Doursat, 1992). The fact that the out-of-sample error is a combination of "true" noise and model mismatch is not limited to DVS plots but should be kept in mind as a necessary limitation of any error-based analysis.

6.3.2 CHARACTERIZATION VIA CONNECTIONIST MODELS. Connectionist models are more flexible than local linear models. We first show how it is possible to extract characteristic properties such as the minimal embedding dimension or the manifold dimension from the network, and then indicate how the network's emulation of the system can be used to estimate Lyapunov coefficients.

For simple feedforward networks (i.e., no direct connections between input and output, and no feedback loops), it is relatively easy to see how the hidden units can be used to discover hidden dimensions:

[36] A comparison to DVS plots for financial data (not shown here) suggests that there are more nonlinearities in the heart than in financial data.

- **Vary network size.** In the early days of backpropagation, networks were trained with varying numbers of hidden units and the "final" test error (when the training "had converged") was plotted as a function of the number of hidden units: it usually first drops and then reaches a minimum; the number of hidden units when the minimum is reached can be viewed as a kind of measure of the degrees of freedom of the system. A similar procedure can determine a kind of embedding dimension by systematically varying the number of input units.
  Problems with this approach are that these numbers can be strongly influenced by the *choice of the activation function* and the search algorithm employed. Ignoring the search issue for the moment: if, for example, sigmoids are chosen as the activation function, we obtain the manifold dimension *as expressed by sigmoids,* which is an upper limit to the true manifold dimension. Saund (1989) suggested, in the context of nonlinear dimensionality reduction, to sandwich the hidden layer (let us now call it the "central" hidden layer) between two (large) additional hidden layers. An interpretation of this architecture is that the time-delay input representation is transformed nonlinearly by the first "encoding" layer of hidden units; if there are more hidden units than inputs, it is an expansion into a higher dimensional space. The goal is to find a representation that makes it easy for the network subsequently to parametrize the manifold with as few parameters as possible (done by the central hidden layer). The prediction is obtained by linearly combining the activations of the final "decoding" hidden layer that follows the central hidden layer.[37]
  Although an expansion with the additional sandwich layers reduces the dependence on the specific choice of the activation function, a small size of the bottleneck layer can make the *search* (via gradient descent in backpropagation) hard: overfitting even occurs for small networks, before they have reached their full potential (Weigend, 1994). There are two approaches to this problem: to penalize network complexity, or to use an oversized network and analyze it.
- **Penalize network complexity or prune.** Most of the algorithms that try to produce small networks have been applied to time series prediction, e.g., "weight elimination" (Weigend, Huberman, & Rumelhart, 1990), "soft

[37] This approach can also be used for cleaning and for compressing time series. In cleaning, we use a filter network that tries to remove noise in the series: the output hopefully corresponds to a noise-reduced version of the signal corresponding to a time at the center of the input time window, rather than a prediction beyond the window. In compression, the network is to reproduce the entire input vector at the output after piping it through a bottleneck layer with a small number of hidden units. The signal at the hidden units is a compressed version of the larger input vector. The three cases of prediction, cleaning, and compression are just different parametrizations of the same manifold, allocating more resources to the areas appropriate for the specific task.

weight sharing" (Nowlan & Hinton, 1992), and "optimal brain damage" (developed by le Cun, Denker, & Solla, 1990, and applied to the time series by Svarer, Hansen, & Larsen, 1993). All of these researchers apply their algorithms to the sunspot series and end up with networks of three hidden units. Finding appropriate parameters for the regularizer can be tricky; we now give a method for dimension estimation that does not have such parameters but does require post-training analysis.

- **Analyze oversized networks.** The idea here is to use a large network that easily reaches the training goal (and also easily overfits).[38] The spectrum of the eigenvalues of the *covariance matrix of the (central) hidden unit activations* is computed as a function of training time. The covariance $C_{ij} = \left\langle (S_i - \overline{S}_i)(S_j - \overline{S}_j) \right\rangle$ describes the two-point interaction between the activations of the two hidden units i and j. ($\overline{S}_i = \langle S_i \rangle$ is the average activation of hidden unit i.) The number of significantly sized eigenvalues of the covariance matrix (its effective rank) serves as a measure of the effective dimension of the hidden unit space (Weigend & Rumelhart, 1991a, 1991b). It expresses the number of parameters needed to parametrize the solution manifold of the dynamical system in terms of the primitives. Using the sandwich-expansion idea (described on the previous page), the effect of the specific primitives can be reduced. We have also used mutual information to capture dependencies between hidden units; this measure is better suited than linear correlation or covariance if the hidden units have nonlinear functional relations.

All of these approaches have to be used with caution as estimates of the true dimension of the manifold. We have pointed out above that the estimate can be too large (for example, if the sigmoid basis functions are not suitable for the manifold, or if the network is overfitting). But it can also be too small (for example, if the network has essentially learned nothing), as often is the case for financial data (either because there is nothing to be emulated or because the training procedure or the architecture was not suited to the data).

In addition to dimension, networks can be used to extract other properties of the generating system. Here we point to a few possibilities that help locate a series within the space of attributes outlined in Figure 2.

Nonlinearity. DVS plots analyze the error as a function of the nonlinearity of the model (smaller neighborhoods ⇒ more nonlinear). Rather than basing the analysis on the errors, we can use a property of the network to characterize the amount of

[38] The network is initialized with very small weights—large enough to break the symmetry but small enough to keep the hidden sigmoids in their linear range. The weights grow as the network learns. In this sense, training time can be viewed in a regularization framework as a complexity term that penalizes weights according their size, strongly at first, and later relaxes.

nonlinearity. Weigend et al. (1990) analyze the distribution of the activations S of sigmoidal hidden units: they show that the ratio of the quadratic part of the Taylor expansion of a sigmoid with respect to the linear part, i.e., $|f''(\xi)|/|f'(\xi)|$, can be expressed in terms of network parameters (the activation S, the net-input ξ, the activation function f, and the slope a are defined in Section 4.2) as $(a|1-2S|)$. The distribution of this statistic (averaged over patterns and hidden units) can be used in addition to the simple comparison of the out-of-sample error of the network to the out-of-sample error of a linear model.

Lyapunov exponents. It is notoriously difficult to estimate Lyapunov exponents from short time records of noisy systems. The hope is that if a network has reliably learned how to emulate such a system, the exponents can be found through the network. This can be done by looking at the out-of-sample errors as a function of prediction time (Weigend et al., 1990), by using the Jacobian of the map implemented by the network (Gencay & Dechert, 1992; Nychka et al., 1992), or by using the trained network to generate time series of any length needed for the application of standard techniques (Brown et al., 1991).

This section on characterization started with important simple tests that apply to any data set and algorithm, continued with redundancy as an example of the detailed information than can be found by analyzing embedded data, and closed with learning algorithms that are more generally applicable but less explicitly understandable. Connectionist approaches to characterization, which throw a broad model (and a lot of computer time) at the data, should be contrasted with the traditional statistical approach of building up nonlinearities by systematically adding terms to a narrow model (which can be estimated much faster than a neural network can be trained) and hoping that the system can be captured by such extensions. This is another example of the central theme of the trade-off that must be made between model flexibility and specificity. The blind (mis)application of these techniques can easily produce meaningless results, but taken together and used thoughtfully, they can yield deep insights into the behavior of a system of which a time series has been observed.

These themes have recurred throughout our survey of new techniques for time series forecasting and characterization. We have seen results that go far beyond what is possible within the canon of linear systems analysis, but we have also seen unprecedented opportunities for the analysis to go astray. We have shown that it can be possible to anticipate, detect, and prevent such errors, and to relate new algorithms to traditional practice, but these steps, as necessary as they are, require significantly more effort. The possibilities that we have presented will hopefully help motivate such an effort. We close this chapter with some thoughts about future directions.

7. THE FUTURE

We have surveyed the results of what appears to be a steady progress of insight over ignorance in analyzing time series. Is there a limit to this development? Can we hope for the discovery of a universal forecasting algorithm that will predict everything about all time series? The answer is emphatically "no!" Even for completely deterministic systems, there are strong bounds on what can be known. The search for a universal time series algorithm is related to Hilbert's vision of reducing all of mathematics to a set of axioms and a decision procedure to test the truth of assertions based on the axioms (*Entscheidungsproblem*); this culminating dream of mathematical research was dramatically dashed by Gödel (1931)[39] and then by Turing. The most familiar result of Turing is the undecidability of the halting problem: it is not possible to decide in advance whether a given computer program will eventually halt (Turing, 1936). But since Turing machines can be implemented with dynamical systems (Fredkin, 1982; Moore, 1991), a universal algorithm that can directly forecast the value of a time series at any future time would need to contain a solution to the halting problem, because it would be able to predict whether a program will eventually halt by examining the program's output. Therefore, there cannot be a universal forecasting algorithm.

The connection between computability and time series goes deeper than this. The invention of Turing machines and the undecidability of the halting problem were side results of Turing's proof of the existence of uncomputable real numbers. Unlike a number such as π, for which there is a rule to calculate successive digits, he showed that there are numbers for which there cannot be a rule to generate their digits. If one was unlucky enough to encounter a deterministic time series generated by a chaotic system with an initial condition that was an uncomputable real number, then the chaotic dynamics would continuously reveal more and more digits of this number. Correctly forecasting the time series would require calculating unseen digits from the observed ones, which is an impossible task.

Perhaps we can be more modest in our aspirations. Instead of seeking complete future knowledge from present observations, a more realistic goal is to find the best model for the data, and a natural definition of "best" is the model that requires the least amount of information to describe it. This is exactly the aim of *Algorithmic Information Theory*, independently developed by Chaitin (1966), Kolmogorov (1965), and Solomonoff (1964). Classical information theory, described in Section 6.2, measures information with respect to a probability distribution of an ensemble of observations of a string of symbols. In algorithmic information theory, information is measured within a single string of symbols by the number of bits needed to specify the shortest algorithm that can generate them. This has led to significant extensions of Gödel and Turing's results (Chaitin, 1990) and, through the *Minimum Description Length* principle, it has been used as the basis for a general theory of statistical inference (Wallace & Boulton, 1968; Rissanen, 1986, 1987). Unfortunately, here again we run afoul of the halting problem. There can be no universal algorithm to find the shortest program to generate an observed sequence because we cannot determine whether an arbitrary candidate program will continue to produce symbols or will halt (e.g., see Cover & Thomas, 1991, p.162).

[39] Hofstadter paraphrases Gödel's Theorem as: "All consistent axiomatic formulations of number theory include undecidable propositions." (Hofstadter, 1979, p. 17).

Although there are deep theoretical limitations on time series analysis, the constraints associated with specific domains of application can nevertheless permit strong useful results (such as those algorithms that performed well in the competition), and can leave room for significant future development. In this chapter, we have ignored many of the most important time series problems that will need to be resolved before the theory can find widespread application, including:

- *Building parametrized models for systems with varying inputs.* For example, we have shown how the stationary dynamics of the laser that produced Data Set A can be correctly inferred from an observed time series, but how can a family of models be built as the laser pump energy is varied in order to gain insight into how this parameter enters into the governing equations? The problem is that for a linear system it is possible to identify an internal transfer function that is independent of the external inputs, but such a separation is not possible for nonlinear systems.
- *Controlling nonlinear systems.* How can observations of a nonlinear system and access to some of its inputs be used to build a model that can then be used to guide the manipulation of the system into a desired state? The control of nonlinear systems has been an area of active research; approaches to this problem include both explicit embedding models (Hübler, 1989; Ott, Grebogi & Yorke, 1990; Bradley, 1992) and implicit connectionist strategies (Miller, Sutton, & Werbos, 1990; White & Sofge, 1992).
- *The analysis of systems that have spatial as well as temporal structure.* The transition to turbulence in a large aspect-ratio convection cell is an experimental example of a spatio-temporal structure (Swinney, this volume), and cellular automata and coupled map lattices have been extensively investigated to explore the theoretical relationship between temporal and spatial ordering (Gutowitz, 1990). A promising approach is to extend time series embedding (in which the task is to find a rule to map past observations to future values) to spatial embedding (which aims to find a map between one spatial, or spatio-temporal, region and another). Unfortunately, the mathematical framework underlying time-delay embedding (such as the uniqueness of state-space trajectories) does not simply carry over to spatial structures. Afraimovich et al. (this volume) explore the prospects for developing such a theory of spatial embedding. A related problem is the combination of data from different sources, ranging from financial problems (using multiple market indicators) to medical data (such as Data Set B).

- *The analysis of nonlinear stochastic processes.* Is it possible to extend embedding to extract a model for the governing equations for a signal generated by a stationary nonlinear stochastic differential equation? Probabilistic approaches to embedding such as the use of expectation values (Eq. (16)) and hidden Markov models (Fraser & Dimitriadis, this volume) may point toward an approach to this problem.
- *Understanding versus learning.* How does understanding (explicitly extracting the geometrical structure of a low-dimensional system) relate to learning (adaptively building models that emulate a complex system)? When a neural network correctly forecasts a low-dimensional system, it has to have formed a representation of the system. What is this representation? Can it be separated from the network's implementation? Can a connection be found between the entrails of the internal structure in a possibly recurrent network, the accessible structure in the state-space reconstruction, the structure in the time series, and ultimately the structure of the underlying system?

The progress in the last decade in analyzing time series has been remarkable and is well witnessed by the contributions to this volume. Where once time series analysis was shaped by linear systems theory, it is now possible to recognize when an apparently complicated time series has been produced by a low-dimensional nonlinear system, to characterize its essential properties, and to build a model that can be used for forecasting. At the opposite extreme, there is now a much richer framework for designing algorithms that can learn the regularities of time series that do not have a simple origin. This progress has been inextricably tied to the arrival of the routine availability of significant computational resources (making it possible to collect large time series, apply complex algorithms, and interactively visualize the results), and it may be expected to continue as the hardware improves.

General access to computer networks has enabled the widespread distribution and collection of common material, a necessary part of the logistics of the competition, thereby identifying interesting results where they once might have been overlooked (such as from students, and from researchers in countries that have only recently been connected to international computer networks). This synchronous style of research is complementary to the more common asynchronous mode of relatively independent publication, and may be expected to become a familiar mechanism for large-scale scientific progress. We hope that, in the short term, the data and results from this competition can continue to serve as basic reference benchmarks for new techniques. We also hope that, in the long term, they will be replaced by more worthy successors and by future comparative studies building on this experience.

In summary, in this overview chapter we started with linear systems theory and saw how ideas from fields such as differential topology, dynamical systems, information theory, and machine learning have helped solve what had appeared to be very difficult problems, leading to both fundamental insights and practical applications. Subject to some ultimate limits, there are grounds to expect significant

further extensions of these approaches for handling a broader range of tasks. We predict that a robust theory of nonlinear time series prediction and analysis (and nonlinear signal processing in general) will emerge that will join spectrum analysis and linear filters in any scientist's working toolkit.

ACKNOWLEDGMENTS

An international interdisciplinary project such as this requires the intellectual and financial support of a range of institutions; we would particularly like to thank the Santa Fe Institute, the NATO Science and Technology Directorate, the Harvard Society of Fellows, the MIT Media Laboratory, the Xerox Palo Alto Research Center, and, most importantly, all of the researchers who contributed to the Santa Fe Time Series Competition.

APPENDIX

We describe the prediction tasks and competition results in more detail. We have included only those entries that were received by the close of the competition; many people have since refined their analyses. This appendix is not intended to serve as an exhaustive catalog of what is possible; it is just a basic guide to what has been done.

DATA SET A: LASER

Submissions were evaluated in two ways. First, using the predicted values $\widehat{x}_k$ only (in addition to the observed values x_k), we compute the *normalized mean squared error*:

$$\mathrm{NMSE}(N) = \frac{\sum_{k\in\mathcal{T}} \left(\mathrm{observation}_k - \mathrm{prediction}_k\right)^2}{\sum_{k\in\mathcal{T}} \left(\mathrm{observation}_k - \mathrm{mean}_{\mathcal{T}}\right)^2} \approx \frac{1}{\widehat{\sigma}^2_{\mathcal{T}}} \frac{1}{N} \sum_{k\in\mathcal{T}} \left(x_k - \widehat{x}_k\right)^2 , \tag{44}$$

where $k = 1 \cdots N$ enumerates the points in the withheld test set $\mathcal{T}$, and $\mathrm{mean}_{\mathcal{T}}$ and $\widehat{\sigma}^2_{\mathcal{T}}$ denote the sample average and sample variance of the observed values (targets) in $\mathcal{T}$. A value of $\mathrm{NMSE} = 1$ corresponds to simply predicting the average.

Second, the submitted error bars $\widehat{\sigma}_k$ were used to compute the likelihood of the observed data, given the predicted values and the predicted error bars, based

TABLE 2 Entries received before the deadline for the prediction of Data Set A (laser). We give the normalized mean squared error (NMSE), and the negative logarithm of the likelihood of the data given the predicted values and predicted errors. Both scores are averaged over the prediction set of 100 points.

code	method	type	computer	time	NMSE(100)	-log(lik.)
W	conn	1-12-12-1; lag 25,5,5	SPARC 2	12 hrs	0.028	3.5
Sa	loc lin	low-pass embd, 8 dim, 4nn	DEC 3100	20 min	0.080	4.8
McL	conn	feedforward, 200-100-1	CRAY Y-MP	3 hrs	0.77	5.5
N	conn	feedforward, 50-20-1	SPARC 1	3 weeks	1.0	6.1
K	visual	look for similar stretches	SG Iris	10 sec	1.5	6.2
L	visual	look for similar stretches			0.45	6.2
M	conn	feedforward, 50-350-50-50	386 PC	5 days	0.38	6.4
Can	conn	recurrent, 4-4c-1	VAX 8530	1 hr	1.4	7.2
U	tree	k-d tree; AIC	VAX 6420	20 min	0.62	7.3
A	loc lin	21 dim, 30 nn	SPARC 2	1 min	0.71	10.
P	loc lin	3 dim time delay	Sun	10 min	1.3	–
Sw	conn	feedforward	SPARC 2	20 hrs	1.5	–
Y	conn	feedforward, weight-decay	SPARC 1	30 min	1.5	–
Car	linear	Wiener filter, width 100	MIPS 3230	30 min	1.9	–

on an assumption of independent Gaussian errors. Although this assumption may not be justified, it provides a simple form that captures some desirable features of error weighting. Since the original real-valued data is quantized to integer values, the probability of seeing a given point x_k is found by integrating the Gaussian distribution over a unit interval (corresponding to the rounding error of 1 bit of the analog-to-digital converter) centered on x_k:

$$p(x_k|\widehat{x}_k,\widehat{\sigma}_k) = \frac{1}{\sqrt{2\pi\widehat{\sigma}_k^2}} \int_{x_k-0.5}^{x_k+0.5} \exp\left(\frac{-(\xi-\widehat{x}_k)^2}{2\widehat{\sigma}_k^2}\right) d\xi \,. \tag{45}$$

If the predicted error is large, then the computed probability will be relatively small, independent of the value of the predicted point. If the predicted error is small and the predicted point is close to the observed value, then the probability will be large, but if the predicted error is small and the prediction is not close to the observed value, then the probability will be very small. The potential reward, as well as the risk, is greater for a confident prediction (small error bars). Under the assumption of independent errors, the likelihood of the whole test for the the observed data given the submitted model is then

$$p(\mathrm{D}|\mathrm{M}) = \prod_{k=1}^{N} p(x_k|\widehat{x}_k,\widehat{\sigma}_k) \,. \tag{46}$$

Finally, we take the logarithm of this probability of the data given the model (this turns products into sums and also avoids numerical problems), and then scale the result by the size of the data set N. This defines the *negative average log likelihood*:

$$-\frac{1}{N}\sum_{k=1}^{N} \log p(x_k|\widehat{x}_k,\widehat{\sigma}_k) \,. \tag{47}$$

We give these two statistics for the submitted entries for Data Set A in Table 2 along with a brief summary of the entries. The best two entries (W=Wan, Sa=Sauer) are shown in Figures 3 and 4 in the main text. Figure 10 displays all predictions received for Data Set A. The symbols ($\times$) correspond to the predicted values, the vertical bars to the submitted error bars. The true continuation points are connected by a grey line (to guide the eye).

DATA SET D: COMPUTER-GENERATED DATA

In order to provide a relatively long series of known high-dimensional dynamics (between the extremes of Data Set A and Data Set C) with weak nonstationarity, we generated 100,000 points by numerically integrating the equations of motion for a damped, driven particle

$$\frac{\mathrm{d}^2\mathbf{x}}{\mathrm{d}t^2} + \gamma\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} + \nabla V(\mathbf{x}) = \mathbf{F}(t) \tag{48}$$

TABLE 3 Entries received before the deadline for the prediction of Data Set D (computer-generated data).

index	method	type	computer	time	NMSE(15)	NMSE(30)	NMSE(50)
ZH	conn	...-30-30-1 & 30-100-5	CM-2 (16k)	8 days	0.086	0.57	0.87
U	tree	k-d tree; AIC	VAX 6420	30 min	1.3	1.4	1.4
C	conn	recurrent, 4-4c-1	VAX 8530	n/a	6.4	3.2	2.2
W	conn	1-30-30-1; lags 20,5,5	SPARC 2	1 day	7.1	3.4	2.4
Z	linear	36 AR(8), last 4k pts.	SPARC	10 min	4.8	5.0	3.2
S	conn	feedforward	SPARC 2	20 hrs	17.	9.5	5.5

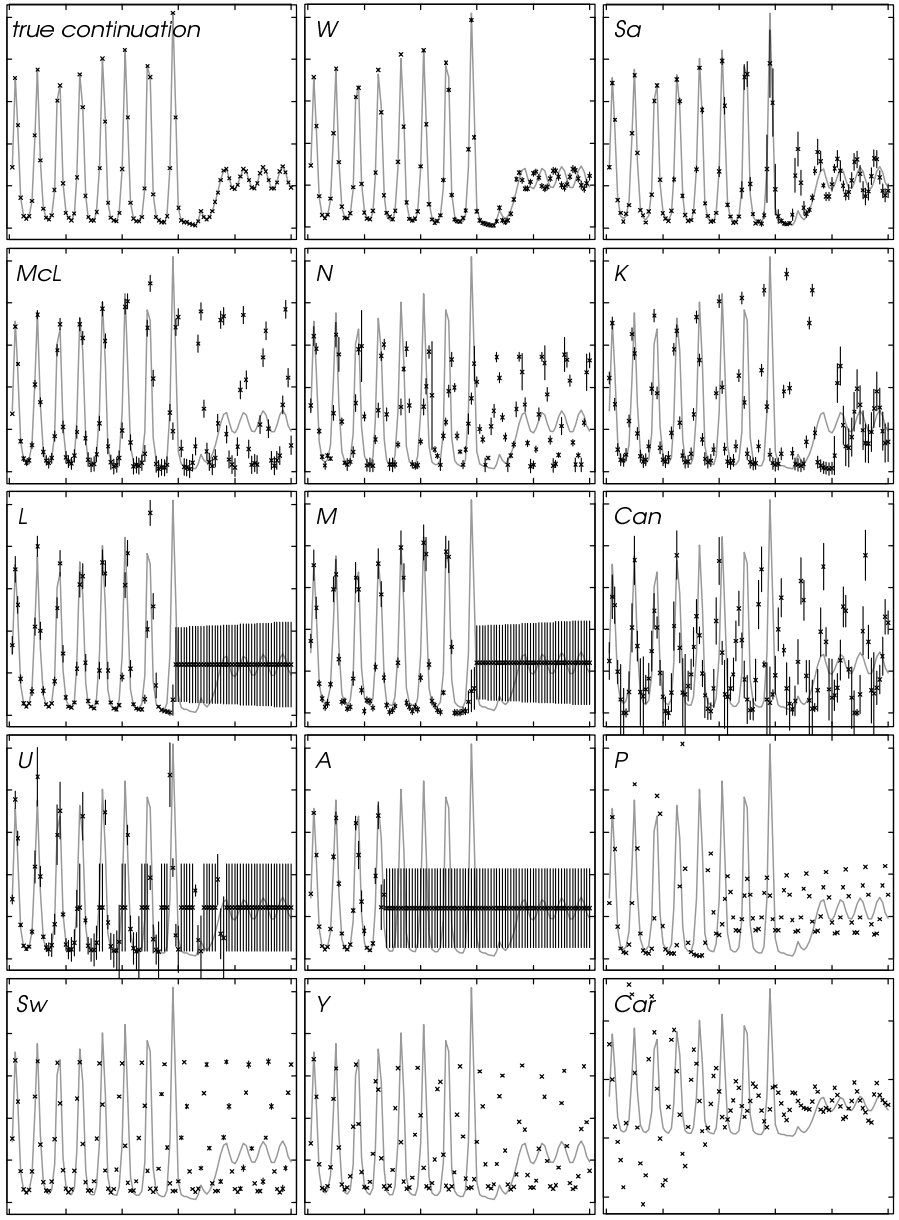

FIGURE 10 Continuations of Data Set A (laser). The letters correspond to the code of the entrant. Grey lines indicate the true continuation, $\times$ the predicted values, and vertical bars the predicted uncertainty.

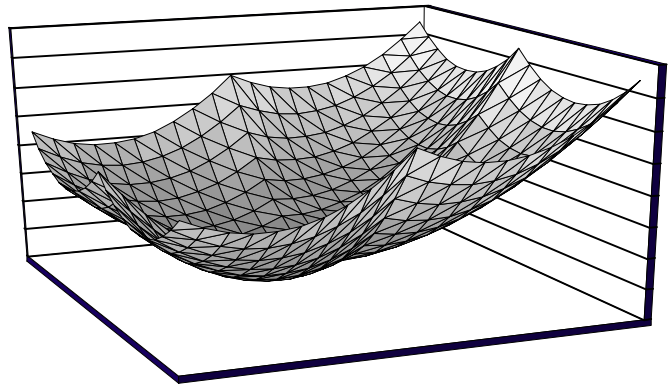

FIGURE 11 The potential $V(\mathbf{x})$ for Data Set D, plotted above the (x_1, x_2) plane.

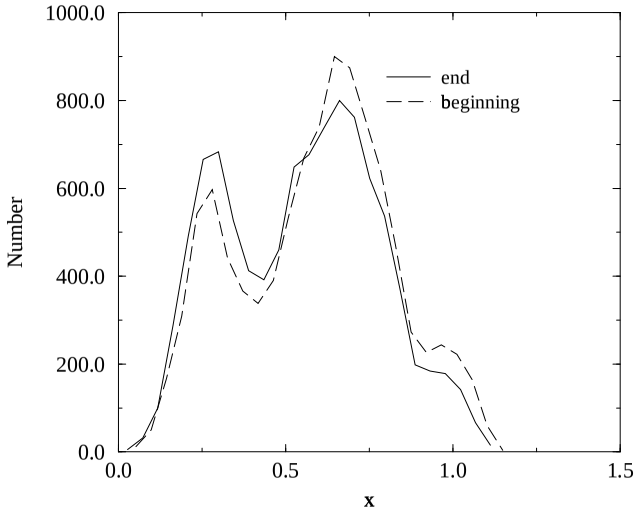

FIGURE 12 Histogram of the probability distribution at the beginning and end of Data Set D, indicating three observable states and the small drift in the relative probabilities.

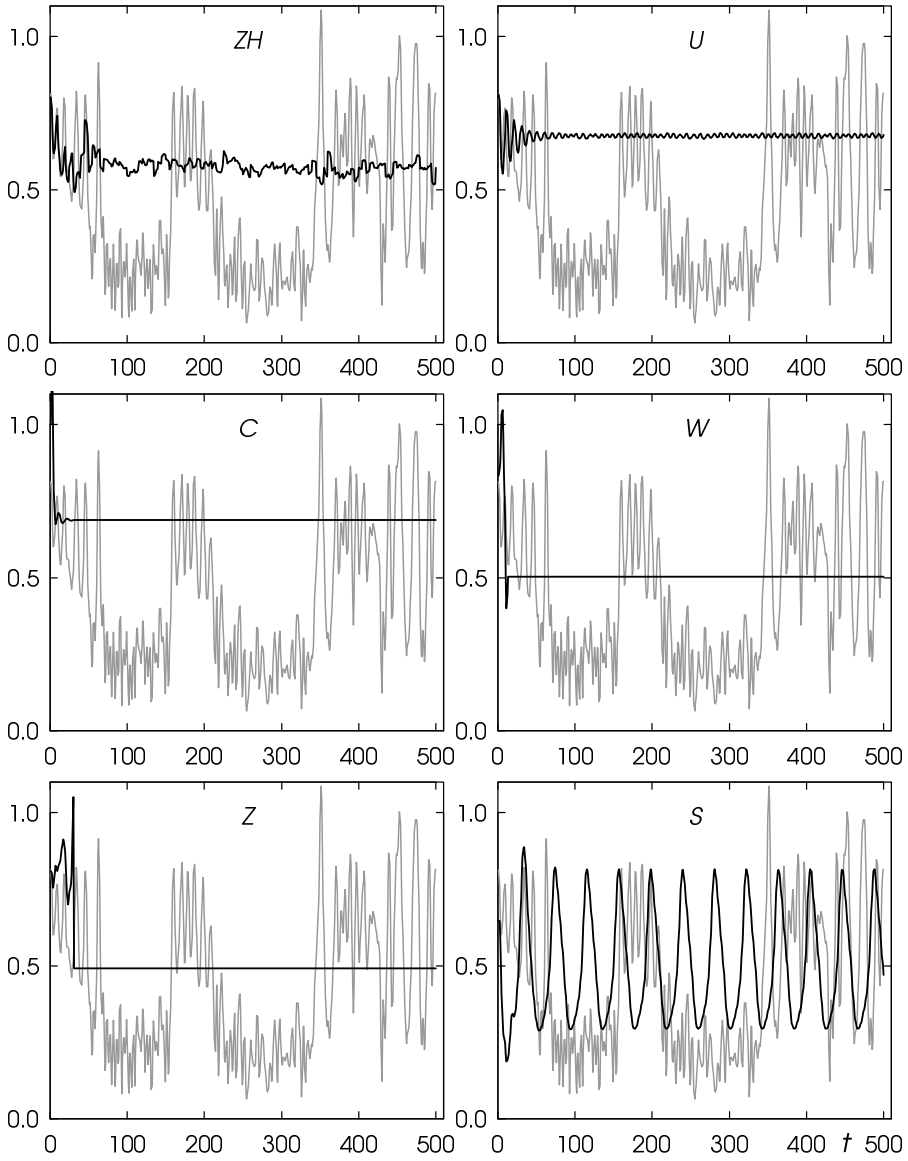

FIGURE 13 Continuations of Data Set D (computer-generated data). The predictions are shown as black lines, the true continuation is indicated in grey.

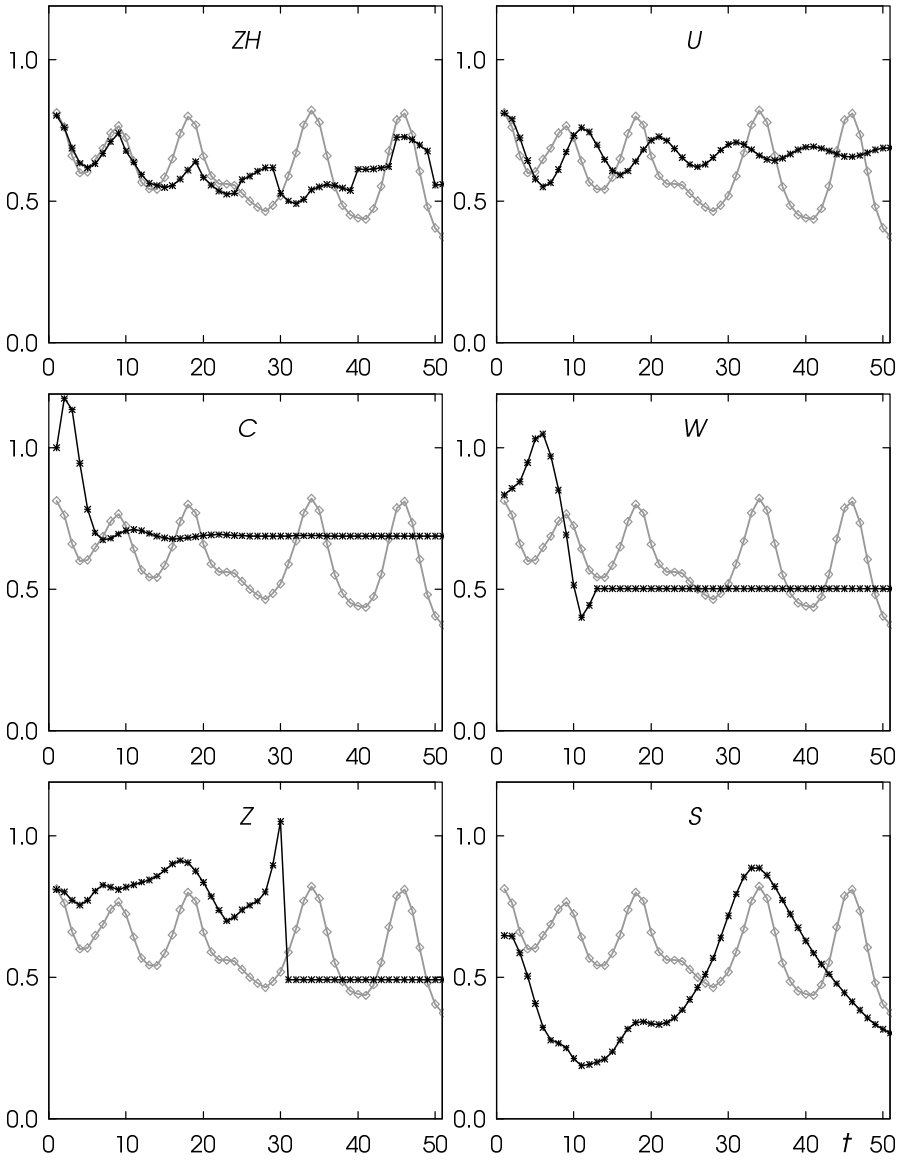

FIGURE 14 Continuations of first 50 points of Data Set D. The predictions are shown as black lines, the true continuation is indicated in grey.

in an asymmetrical four-dimensional four-well potential (Figure 11)

$$V(\mathbf{x}) \;=\; A_4 \left(x_1^2 + x_2^2 + x_3^2 + x_4^2\right)^2 - A_2 \;|x_1 x_2| - A_1 x_1 \tag{49}$$

with periodic forcing

$$\mathbf{F}(t) = F \sin(\omega t)\; \widehat{\mathbf{x}}_3 \,. \tag{50}$$

$\mathbf{x} = (x_1, x_2, x_3, x_4)$ denotes the location of the particle. This system has nine degrees of freedom (four position, four velocity, and one forcing time). The equations were integrated with 64-bit real numbers using a fixed-step fourth-order Runge-Kutta algorithm (to eliminate possible coupling to an active stepper). The potential has four wells that are tilted by the parameter A_1. This parameter slowly drifted during the generation of the data according to a small biased random walk from 0.02 to 0.06. As a scalar observable we chose

$$\sqrt{(x_1 - 0.3)^2 + (x_2 - 0.3)^2 + x_3^2 + x_4^2}\,. \tag{51}$$

This observable projects the four wells onto three distinguishable states. These three states, and the effect of the drift of A_1 can be seen in the probability distributions at the beginning and the end of the data set (Figure 12). The magnitude of the drift is chosen to be small enough such that the nature of the dynamics does not change, but large enough such that the relative probabilities are different at the beginning and end of the data set. Data Set D was generated with $A_4 = 1$, $A_2 = 1$, A_1 drifting from 0.02 to 0.06, $\gamma = 0.01$, $F = 0.135$, and with $\omega = 0.6$ (for these parameters, in the absence of the weak forcing by the drift, this system is not chaotic).

The contributions received before the deadline are listed in Table 3; they are evaluated by normalized mean squared error, NMSE, averaged over the first 15, 30, and 50 predictions. Figure 13 shows the predictions received for this data set over the entire prediction range of 500 points. The first 50 points of the prediction range are plotted again in Figure 14. The first 15 steps of the best entry (ZH = Zhang and Hutchinson) is also shown in Figure 7 in the main text, in addition to the prediction of the evolution of the probability density function, submitted by Fraser and Dimitriadis after the deadline of the competition. On this time scale, the spread of an ensemble of continuations due to the stochasticity of the algorithm used to generate Dat Set D is small ($\sim 1\%$).

The program and parameter file that we used to generate Data Set D are available through anonymous ftp to `ftp.santafe.edu` .

Appendix to the Book: Accessing the Server

The data used for the *Santa Fe Time Series Prediction and Analysis Competition,* programs, results of analyses, visualizations, sonifications, as well as other data sets, are available from the **Time-Series** archive at **ftp.santafe.edu**. Retrieving and depositing material is straightforward by ftp over the Internet. In the following example session, we show **user input** in bold, `system response` (some parts are deleted here) in teletype font, and *comments* in italics. If you cannot resolve problems with ftp, send e-mail to **ftp@santafe.edu** .

From your computer type:

```
rak_pop% ftp ftp.santafe.edu
Name:    anonymous
Password: your_email_address
ftp>  cd pub/Time-Series
ftp>  dir
-rw-rw-r-- 1 tserver 113 3522 Jun 12 1992 README
```

This file describes the available directories, and it contains the necessary forms to submit data sets and programs to the archive.

```
drwxrwxr-x 2 tserver 113 512 Jun 12 1992 competition
```

This directory has the competition data sets (along with their description) and the original instructions for the competition.

```
drwxrwxr-x 2 tserver 113 512 Jul 28 1750 Bach
```
This directory contains the data of the Bach fugue and its continuations. See Dirst and Weigend (this volume).

```
drwxrwxr-x 2 tserver 113 512 Jun 12 1992 results
```
This directory is for comparative evaluations of time series techniques and data sets.

```
drwxrwxr-x 2 tserver 113 512 Jun 12 1992 programs
```
This directory is for programs related to time series analysis.

```
drwxrwxr-x 2 tserver 113 512 Nov 16 1992 data
```
This directory contains data sets that have been added following the close of the competition, as well as some standard data sets such as the sunspot series. Please note that some of the data sets are large (e.g., some of the financial data sets are more than 3MB compressed). Please transfer large files only at night.

```
drwxrwx-wx 3 tserver 113 512 Apr 13 05:17 incoming
```
New material should be deposited here and timeseries@santafe.edu notified.

```
ftp> cd competition
ftp> dir
-rw-rw-r--  1 tserver 113     54558 Feb  1  1992 A.cont
-rw-rw-r--  1 tserver 113      6000 Aug  3  1991 A.dat
-rw-rw-r--  1 tserver 113    291238 Aug  3  1991 B1.dat
-rw-rw-r--  1 tserver 113    295321 Aug  3  1991 B2.dat
-rw-rw-r--  1 tserver 113      1778 Feb  1  1992 C.cont
-rw-rw-r--  1 tserver 113    285090 Aug  1  1991 C1-5.dat
-rw-rw-r--  1 tserver 113    285091 Aug  1  1991 C6-10.dat
-rw-rw-r--  1 tserver 113      3000 Feb  1  1992 D.cont
-rw-rw-r--  1 tserver 113    300000 Aug  3  1991 D1.dat
-rw-rw-r--  1 tserver 113    300000 Aug  3  1991 D2.dat
-rw-rw-r--  1 tserver 113    217946 Aug  3  1991 E.dat
-rw-rw-r--  1 tserver 113     55687 Dec 29  1991 F.dat
-rw-rw-r--  1 tserver 113     21196 Feb  1  1992 data_information
-rw-rw-r--  1 tserver 113     29231 Dec 30  1991 old_instructions
ftp> get A.dat
150 Opening ASCII mode data connect for A.dat (6000 bytes).
226 Transfer complete.
ftp> bye
```

References

We here list the references for this chapter (Gershenfeld & Weigend, 1993). They are extracted from the full bibliography of the book.

Abelson, H. 1990. "The Bifurcation Interpreter: A Step Towards the Automatic Analysis of Dynamical Systems." *Intl. J. Comp. & Math. Appl.* **20**:13.

Afraimovich, V. S., M. I. Rabinovich, and A. L. Zheleznyak. 1993. "Finite-Dimensional Spatial Disorder: Description and Analysis." In *Time Series Prediction: Forecasting the Future and Understanding the Past*, edited by A. S. Weigend and N. A. Gershenfeld, 539–556. Reading, MA: Addison-Wesley.

Akaike, H. 1970. "Statistical Predictor Identification." *Ann. Inst. Stat. Math.* **22**: 203–217.

Alesic, Z. 1991. "Estimating the Embedding Dimension." *Physica D* **52**: 362–368.

Barron, A. R. 1993. "Universal Approximation Bounds for Superpositions of a Sigmoidal Function." *IEEE Trans. Info. Theory* **39(3)**: 930–945.

Beck, C. 1990. "Upper and Lower Bounds on the Renyi Dimensions and the Uniformity of Multifractals." *Physica D* **41**: 67–78.

Bourgoin, M., K. Sims, S. J. Smith, and H. Voorhees. 1993. "Learning Image Classification with Simple Systems and Large Databases." *IEEE Trans. Pat. Anal. & Mach. Intel.*: submitted.

Box, G. E. P., and F. M. Jenkins. 1976. *Time Series Analysis: Forecasting and Control*, 2nd ed. Oakland, CA: Holden-Day.

Bradley, E. 1992. "Taming Chaotic Circuits." Ph.D. Thesis, Massachusetts Institute of Technology, September 1992.

Brock, W. A., W. D. Dechert, J. A. Scheinkman, and B. LeBaron. 1988. "A Test For Independence Based on the Correlation Dimension." University of Wisconsin Press, Madison, WI.

Broomhead, D. S., and G. P. King. 1986. "Extracting Qualitative Dynamics from Experimental Data." *Physica D* **20**: 217–236.

Broomhead, D. S., and D. Lowe. 1988. "Multivariable Functional Interpolation and Adaptive Networks." *Complex Systems* **2**: 321–355.

Brown, R., P. Bryant, and H. D. I. Abarbanel. 1991. "Computing the Lyapunov Spectrum of a Dynamical System from an Observed Time Series." *Phys. Rev. A* **43**: 2787–806.

Buntine, W. L., and A. S. Weigend. 1991. "Bayesian Backpropagation." *Complex Systems* **5**: 603–643.

Casdagli, M. 1989. "Nonlinear Prediction of Chaotic Time Series." *Physica D* **35**: 335–356.

Casdagli, M. 1991. "Chaos and Deterministic versus Stochastic Nonlinear Modeling." *J. Roy. Stat. Soc. B* **54**: 303–328.

Casdagli, M., S. Eubank, J. D. Farmer, and J. Gibson. 1991. "State Space Reconstruction in the Presence of Noise." *Physica D* **51D**: 52–98.

Casdagli, M. C., and A. S. Weigend. 1993. "Exploring the Continuum Between Deterministic and Stochastic Modeling." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 347-366. Reading, MA: Addison-Wesley.

Catlin, D. E. 1989. *Estimation, Control, and the Discrete Kalman Filter*. Applied Mathematical Sciences, Vol. 71. New York: Springer-Verlag, 1989.

Chaitin, G. J. 1966. "On the Length of Programs for Computing Finite Binary Sequences." *J. Assoc. Comp. Mach* **13**: 547–569.

Chaitin, G. J. 1990. *Information, Randomness & Incompleteness*. Series in Computer Science, Vol. 8, 2nd ed. Singapore: World-Scientific.

Chatfield, C. 1988. "What is the Best Method in Forecasting?" *J. Appl. Stat.* **15**: 19–38.

Chatfield, C. 1989. *The Analysis of Time Series*, 4th ed. London: Chapman and Hall, 1989.

Clemens, C. 1993. "Whole Earth Telescope Observations of the White Dwarf Star (PG1159-035)." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 139–150. Reading, MA: Addison-Wesley.

Collet, P., and J.-P. Eckmann. 1980. *Iterated Maps on the Interval as Dynamical Systems*. Boston: Birkhäuser.

Cover, T. M. 1965. "Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition." *IEEE Trans. Elec. Comp.* **14**: 326–334.

Cover, T. M., and J. A. Thomas. 1991. *Elements of Information Theory* . New York: John Wiley.

Cremers, J., and A. Hübler. 1987. "Construction of Differential Equations from Experimental Data." *Z. Naturforsch.* **42(a)**: 797–802.

Crutchfield, J. P., and B. S. McNamara. 1987. "Equations of Motion from a Data Series." *Complex Systems* **1**: 417–452.

Crutchfield, J. P., and K. Young. 1989. "Inferring Statistical Complexity." *Phys. Rev. Lett.* **63**: 105–108.

Cybenko, G. 1989. "Approximation by Superpositions of a Sigmoidal Function." *Math. Control, Signals, & Sys.* **2(4)**.

Diebold, F. X., and J. M. Nason. 1990. "Nonparametric Exchange Rate Prediction?" *J. Intl. Econ.* **28**: 315–332.

Dirst, M., and A. S. Weigend. 1993. "Baroque Forecasting: On Completing J. S. Bach's Last Fugue." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 151–172. Reading, MA: Addison-Wesley.

Duda, R. O., and P. E. Hart. 1973. *Pattern Classification and Scene Analysis*. New York: Wiley.

Dutta, P., and P. M. Horn. 1981. "Low-Frequency Fluctuations in solids—1/f Noise." *Rev. Mod. Phys.* **53**: 497–516.

Farmer, J. D., and J. J. Sidorowich. 1987. "Predicting Chaotic Time Series." *Phys. Rev. Lett.* **59(8)**: 845–848.

Farmer, J. D., and J. J. Sidorowich. 1988. "Exploiting Chaos to Predict the Future and Reduce Noise." *Evolution, Learning, and Cognition*, edited by Y. C. Lee. Singapore: World Scientific.

Fraser, A. M. 1989a. "Reconstructing Attractors from Scalar Time Series: A Comparison of Singular System and Redundancy Criteria." *Physica D* **34**: 391–404.

Fraser, A. M. 1989b. "Information and Entropy in Strange Attractors." *IEEE Trans. Info. Theory* **IT-35**: 245–262.

Fraser, A. M. 1989c. "Reconstructing Attractors from Scalar Time Series: A Comparison of Singular System and Redundancy Criteria." *Physica D* **34**: 391–404.

Fraser, A. M., and A. Dimitriadis. 1993. "Forecasting Probability Densities Using Hidden Markov Models with Mixed States." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 265–282. Reading, MA: Addison-Wesley.

Fraser, A. M., and H. L. Swinney. 1986. "Independent Coordinates for Strange Attractors from Mutual Information." *Phys. Rev. A* **33**: 1134–1140.

Fredkin, E., and T. Toffoli. 1982. "Conservative Logic." *Intl. J. Theor. Phys.* **21**: 219–253.

Friedman, J. H. 1991c. "Multivariate Adaptive Regression Splines." *Ann. Stat.* **19**: 1–142. With discussion.

Funahashi, K.-I. 1989. "On the Approximate Realization of Continuous Mappings by Neural Networks." *Neur. Net.* **2**: 183–192.

Geman, S., E. Bienenstock, and R. Doursat. 1992. "Neural Networks and the Bias/Variance Dilemma." *Neur. Comp.* **5**: 1–58.

Gencay, R., and W. D. Dechert. 1992. "An Algorithm for the n Lyapunov Exponents of an n-Dimensional Unknown Dynamical System." *Physics D* **59**:142–157.

Gershenfeld, N. A. 1989. "An Experimentalist's Introduction to the Observation of Dynamical Systems." In *Directions in Chaos*, edited by B.-L. Hao, Vol. 2, 310–384. Singapore: World Scientific.

Gershenfeld, N. A. 1992. "Dimension Measurement on High-Dimensional Systems." *Physica D* **55**: 135–154.

Gershenfeld, N. A. 1993a. "Embedding, Expectations, and Noise." Preprint.

Gershenfeld, N. A. 1993b. "Information in Dynamics." In *Proceedings of the Workshop on Physics of Computation*, edited by D. Matzke, 276–280. Los Alamitos, CA: IEEE Press.

Gingerich, O. 1992. *The Great Copernicus Chase and Other Adventures in Astronomical History.* Cambridge, MA: Sky.

Giona, M., F. Lentini, and V. Cimagalli. 1991. "Functional Reconstruction and Local Prediction of Chaotic Time Series." *Phys. Rev. A* **44**: 3496–3502.

Gödel, K. 1931. "Über formal unentscheibare Sätze der *Principia Mathematica* und verwandter Systeme, I." *Monatshefte für Mathematik und Physik* **38**: 173–198. An English translation of this paper is found in *On Formally Undecidable Propositions* by K. Gödel (New York: Basic Books, 1962).

Granger, C. W. J., and A. P. Andersen. 1978. *An Introduction to Bilinear Time Series Models.* Gottingen: Vandenhoek and Ruprecht.

Grassberger, P. 1988. "Finite Sample Corrections to Entropy and Dimension Estimates." *Phys. Lett A* **128**: 369–373.

Grassberger, P., and I. Procaccia. 1983a. "Characterization of Strange Attractors." *Phys. Rev. Lett.* **50**: 346–349.

Grebogi, C., S. M. Hammel, J. A. Yorke, and T. Sauer. 1990. "Shadowing of Physical Trajectories in Chaotic Dynamics: Containment and Refinement." *Phys. Rev. Lett.* **65**: 1527.

Green, M. L., and R. Savit. 1991. "Dependent Variables in Broadband Continuous Time Series." *Physica D* **50**: 521–544.

Guckenheimer, J. 1982. "Noise in Chaotic Systems." *Nature* **298**: 358–361.

Guillemin, V., and A. Pollack. 1974. *Differential Topology.* Englewood Cliffs, NJ: Prentice-Hall.

Gutowitz, H., ed. 1991. *Cellular Automata, Theory and Experiment.* Cambridge, MA: MIT Press.

Hammerstrom, D. 1993. "Neural Networks at Work." *IEEE Spectrum* June: 26–32.

Hentschel, H. G. E., and I. Procaccia. 1983. "The Infinite Number of Generalized Dimensions of Fractals and Strange Attractors." *Physica D* **8**: 435–444.

Hertz, J. A., A. S. Krogh, and R. G. Palmer. 1991. *Introduction to the Theory of Neural Computation.* Santa Fe Institute Studies in the Sciences of Complexity, Lect. Notes Vol. I. Redwood City, CA: Addison-Wesley.

Hinton, G. E., and T. J. Sejnowski. 1986. "Learning and Relearning in Boltzmann Machines." In *Parallel Distributed Processing*, edited by D. E. Rumelhart and J. L. McClelland, volume 1. Cambridge, MA: MIT Press.

Hinton, G. E., and D. van Camp. 1993. "Keeping Neural Networks Simple by Minimizing the Description Length of the Weights." Preprint, Computer Science Department, University of Toronto, June 1993.

Hofstadter, D. R. 1979. *Gödel, Escher, Bach: An Eternal Golden Braid.* New York: Basic Books.

Hu, M. J. C. 1964. "Application of the Adaline System to Weather Forecasting." E. E. Degree Thesis. Technical Report 6775-1, Stanford Electronic Laboratories, Stanford, CA, June.

Hübler, A. 1989. "Adaptive Control of Chaotic Systems." *Helv. Phys. Acta* **62**: 343–346.

Hübner, U., C. O. Weiss, N. B. Abraham, and D. Tang. 1993. "Lorenz-Like Chaos in NH_3-FIR Lasers." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 73-104. Reading, MA: Addison-Wesley.

Irie, B., and S. Miyake. 1988. "Capabilities of Three-Layered Perceptrons." *Proceedings of the IEEE International Conference on Neural Networks, San Diego*, I: 641–648.

Kantz, H. 1993. "Noise Reduction by Local Reconstruction of the Dynamics." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 475–490. Reading, MA: Addison-Wesley.

Kaplan, D. T. 1993. "A Geometrical Statistic for Detecting Deterministic Dynamics." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 415–428. Reading, MA: Addison-Wesley.

Kennel, M. B., R. Brown, and H. D. I. Abarbanel. 1992. "Determining Minimum Embedding Dimension Using a Geometrical Construction." *Phys. Rev. A* **45**: 3403–3411.

Knuth, D. E. 1981. *Semi-Numerical Algorithms. Art of Computer Programming*, Vol. 2, 2nd ed. Reading, MA: Addison-Wesley.

Kolmogorov, A. 1941. "Interpolation und Extrapolation von stationären zufälligen Folgen." *Bull. Acad. Sci. (Nauk)* **5**: 3–14. U.S.S.R., Ser. Math.

Kolmogorov, A. N. 1965. "Three Approaches to the Quantitative Definition of Information." *Prob. Infor. Trans.* **1**: 4–7.

Koza, J. R. 1993. *Genetic Programming*. Cambridge, MA: MIT Press.

Kung, S. Y. 1993. *Digital Neural Networks*. Englewood Cliffs, NJ: Prentice Hall.

Laird, P., and R. Saul. 1993. "Discrete Sequence Prediction and Its Applications." *Machine Learning*: submitted.

Landauer, R. 1991. "Information is Physical." *Physics Today* **44**: 23.

Lang, K. J., A. H. Waibel, and G. E. Hinton. 1990. "A Time-Delay Neural Network Architecture for Isolated Word Recognition." *Neur. Net.* **3**: 23–43.

Lapedes, A., and R. Farber. 1987. "Nonlinear Signal Processing Using Neural Networks." Technical Report No. LA-UR-87-2662, Los Alamos National Laboratory, Los Alamos, NM.

LeBaron, B. 1993. "Nonlinear Diagnostics and Simple Trading Rules for High-Frequency Foreign Exchange Rates." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 457–474. Reading, MA: Addison-Wesley.

le Cun, Y. 1989. "Generalization and Network Design Strategies." In *Connectionism in Perspective*, edited by R. Pfeifer, Z. Schreter, F. Fogelman, and L. Steels. Amsterdam: North Holland.

le Cun, Y. J. S. Denker, and S. A. Solla. 1990. "Optimal Brain Damage." In *Advances in Neural Information Processing Systems 2 (NIPS*89)*, edited by D. S. Touretzky, 598–605. San Mateo, CA: Morgan Kaufmann.

Lequarré, J. Y. 1993. "Foreign Currency Dealing: A Brief Introduction." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 131-137. Reading, MA: Addison-Wesley.

Lewis, P. A. W., B. K. Ray, and J. G. Stevens. 1993. "Modeling Time Series Using Multivariate Adaptive Regression Splines (MARS)." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 296-318. Reading, MA: Addison-Wesley.

Liebert, W., and H. G. Schuster. 1989. "Proper Choice of the Time Delay for the Analysis of Chaotic Time Series." *Phys. Lett. A* **142**: 107–111.

Lorenz, E. N. 1963. "Deterministic Non-Periodic Flow." *J. Atmos. Sci.* **20**: 130–141.

Lorenz, E. N. 1989. "Computational Chaos—A Prelude to Computational Instability." *Physica D* **35**: 299–317.

Makridakis, S., A. Andersen, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, and R. Winkler. 1984. *The Forecasting Accuracy of Major Time Series Methods*. New York: Wiley.

Makridakis, S., and M. Hibon. 1979. "Accuracy of Forecasting: An Empirical Investigation." *J. Roy. Stat. Soc. A* **142**: 97–145. With discussion.

Marteau, P. F., and H. D. I. Abarbanel. 1991. "Noise Reduction in Chaotic Time Series Using Scaled Probabilistic Methods." *J. Nonlinear Sci.* **1**: 313.

May, R. M. 1976. "Simple Mathematical Models with Very Complicated Dynamics." *Nature* **261**: 459.

Melvin, P., and N. B. Tufillaro. 1991. "Templates and Framed Braids." *Phys. Rev. A* **44**: R3419–R3422.

Meyer, T. P., and N. H. Packard. 1992. "Local Forecasting of High-Dimensional Chaotic Dynamics." In *Nonlinear Modeling and Forecasting*, edited by M. Casdagli and S. Eubank. Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XII, 249–263. Reading, MA: Addison-Wesley.

Miller, W. T., R. S. Sutton, and P. J. Werbos. 1990. *Neural Networks for Control*. Cambridge, MA: MIT Press.

Mitchison, G. J., and R. M. Durbin. 1989. "Bounds on the Learning Capacity of Some Multi-Layer Networks." *Biol. Cyber.* **60**: 345–356.

Moody, J. 1992. "The *Effective* Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Systems." In *Advances in Neural Information Processing Systems 4*, edited by J. E. Moody, S. J. Hanson, and R. P. Lippmann. San Mateo, CA: Morgan Kaufmann.

Moore, C. 1991. "Generalized Shifts: Unpredictability and Undecidability in Dynamical Systems." *Nonlinearity* **4**: 199–230.

Mozer, M. C. 1993. "Neural Net Architectures for Temporal Sequence Processing." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 243–264. Reading, MA: Addison-Wesley.

Nowlan, S. J., and G. E. Hinton. 1992. "Simplifying Neural Networks by Soft Weight-Sharing." *Neur. Comp.* **4**: 473–493.

Nychka, D., S. Ellner, D. McCaffrey, and A. R. Gallant. 1992. "Finding Chaos in Noisy Systems." *J. Roy. Stat. Soc. B* **54(2)**: 399–426.

Oppenheim, A. V., and R. W. Schafer. 1989. *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice Hall.

Ott, E., C. Grebogi, and J. A. Yorke. 1990. "Controlling Chaos." *Phys. Rev. Lett.* **64**: 1196.

Packard, N. H. 1990. "A Genetic Learning Algorithm for the Analysis of Complex Data." *Complex Systems* **4**: 543–572.

Packard, N. H., J. P. Crutchfeld, J. D. Farmer, and R. S. Shaw. 1980. "Geometry from a Time Series." *Phys. Rev. Lett.* **45(9)**: 712–716.

Paluš, M. 1993b. "Identifying and Quantifying Chaos Using Information-Theoretic Functionals." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 387–413. Reading, MA: Addison-Wesley.

Parlitz, U. 1992. "Identification of True and Spurious Lyapunov Exponents from Time Series." *Intl. J. Bif. & Chaos* **2**: 155–165.

Petersen, K. 1989. *Ergodic Theory*, 2nd ed. Cambridge Studies in Advanced Mathematics, Vol. 2. Cambridge, MA: Cambridge University Press.

Pettis, K. W., T. A. Bailey, A. K. Jain, and R. C. Dubes. 1979. "An Intrinsic Dimensionality Estimator from Near-Neighbor Information." *IEEE Trans. Patt. Anal. & Mach. Intel.* **PAMI-1**: 25–37.

Pi, H., and C. Peterson. 1993. "Finding the Embedding Dimension and Variable Dependences in Time Series." Preprint LU TP 93-4, Department of Theoretical Physics, University of Lund, March 1993. Submitted to *Neural Computation.*

Pineda, F. J., and J. C. Sommerer. 1993. "Estimating Generalized Dimensions and Choosing Time Delays: A Fast Algorithm." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 367–385. Reading, MA: Addison-Wesley.

Poggio, T., and F. Girosi. 1990. "Networks for Approximation and Learning." *Proc. IEEE* **78(9)**: 1481–1497.

Powell, M. J. D. 1987. "Radial Basis Functions for Multivariate Interpolation: A Review." In *IMA Conference on "Algorithms for the Approximation of Functions and Data,"* edited by J. C. Mason and M. G. Cox. Shrivenham: RMCS.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. 1992. *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge: Cambridge University Press.

Priestley, M. 1981. *Spectral Analysis and Time Series*. London: Academic Press.

Principe, J. C., B. de Vries, and P. Oliveira. 1993. "The Gamma Filter—A New Class of Adaptive IIR Filters with Restricted Feedback." *IEEE Trans. Sig. Proc.* **41**: 649–656.

Rabiner, L. R., and B. H. Juang. 1986. "An Introduction to Hidden Markov Models." *IEEE ASSP Magazine*, January: 4–16.

Rico-Martinez, R., I. G. Kevrekidis, and R. A. Adomaitis. 1993. "Noninvertibility in Neural Networks." In *Proceedings of ICNN, San Francisco, 1993*, 382–386. Piscataway, NJ: IEEE Press.

Rigney, D. R., A. L. Goldberger, W. C. Ocasio, Y. Ichimaru, G. B. Moody, and R. G. Mark. 1993. "Multi-Channel Physiological Data: Description and Analysis." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 105–129. Reading, MA: Addison-Wesley.

Rissanen, J. 1986. "Stochastic Complexity and Modeling." *Ann. Stat.* **14**: 1080–1100.

Rissanen, J. 1987. "Stochastic Complexity." *J. Roy. Stat. Soc. B* **49**: 223–239. With discussion: 252–265.

Rissanen, J., and G. G. Langdon. 1981. "Universal Modeling and Coding." *IEEE Trans. Info. Theory.* **IT-27**: 12–23.

Ruelle, D., and J. P. Eckmann. 1985. "Ergodic Theory of Chaos and Strange Attractors." *Rev. Mod. Phys.* **57**: 617–656.

Rumelhart, D. E., R. Durbin, R. Golden, and Y. Chauvin. 1993. "Backpropagation: The Basic Theory." In *Backpropagation: Theory, Architectures and Applications*, edited by Y. Chauvin and D. E. Rumelhart. Hillsdale, NJ: Lawrence Erlbaum.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams. 1986a. "Learning Internal Representations by Error Propagation." In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume I: Foundations*, edited by D. E. Rumelhart and J. L. McClelland, 318–362. Cambridge, MA: MIT Press/Bradford Books.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams. 1986b. "Learning Representations by Back-Propagating Errors." *Nature* **323**: 533–536.

Sakamoto, Y., M. Ishiguro, and G. Kitagawa. 1986. *Akaike Information Criterion Statistics*. Dordrecht: D. Reidel.

Sauer, T. 1992. "A Noise Reduction Method for Signals from Nonlinear Systems." *Physica D* **58**: 193–201.

Sauer, T. 1993. "Time Series Prediction Using Delay Coordinate Embedding." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 175–193. Reading, MA: Addison-Wesley.

Sauer, T., J. A. Yorke, and M. Casdagli. 1991. "Embedology." *J. Stat. Phys.* **65(3/4)**: 579–616.

Saund, E. 1989. "Dimensionality-Reduction Using Connectionist Networks." *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)* **11**: 304–314.

Schuster, A. 1898. "On the Investigation of Hidden Periodicities with Applications to a Supposed 26-Day Period of Meteorological Phenomena." *Terr. Mag.* **3**: 13–41.

Schwartz, E. I. 1992. "Where Neural Networks are Already at Work: Putting AI to Work in the Markets." *Bus. Week* November 2: 136–137.

Shannon, C. E. 1948. "A Mathematical Theory of Communication." *Bell Syst. Tech. J.* **27**: 379–423, 623–656. Reprinted in *Key Papers in the Development of Information Theory,* edited by D. Slepian, 5–18. New York: IEEE Press.

Shaw, R. S. 1981. "Strange Attractors, Chaotic Behavior and Information Flow." *Z. Naturforsch.* **36A**: 80–112.

Smith, L. A. 1993. "Does a Meeting in Sante Fe Imply Chaos?" In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 323-343. Reading, MA: Addison-Wesley.

Smolensky, P., M. C. Mozer, and D. E. Rumelhart, eds. 1994. *Mathematical Perspectives on Neural Networks*. Hillsdale, NJ: Lawrence Erlbaum.

Solomonoff, R. J. 1964. "A Formal Theory of Induction Inference, Parts I and II." *Information & Control* **7**: 1–22, 221–254.

Subba Rao, T. 1992. "Analysis of Nonlinear Time Series (and Chaos) by Bispectral Methods." In *Nonlinear Modeling and Forecasting*, edited by M. Casdagli and S. Eubank. Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XII, 199–226. Reading, MA: Addison-Wesley.

Sussman, G. J., and J. Wisdom. 1988. "Numerical Evidence that the Motion of Pluto is Chaotic." *Science* **241**: 433–437.

Sussman, G. J., and J. Wisdom. 1992. "Chaotic Evolution of the Solar System." *Science* **257** (1992): 56–62.

Svarer, C., L. K. Hansen, and J. Larsen. 1993. "On Design and Evaluation of Tapped-Delay Neural Network Architectures." In *IEEE International Conference on Neural Networks, San Francisco (March 1993)*, 46–51. Piscataway, NJ: IEEE Service Center.

Swinney, H. L. 1993. "Spatio-Temporal Patterns: Observations and Analysis." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 557–567. Reading, MA: Addison-Wesley.

Takens, F. 1981. "Detecting Strange Attractors in Turbulence." In *Dynamical Systems and Turbulence*, edited by D. A. Rand and L.-S. Young. Lecture Notes in Mathematics, Vol. 898, 336–381. Warwick, 1980. Berlin: Springer-Verlag.

Temam, R. 1988. *Infinite-Dimensional Dynamical Systems in Mechanics and Physics*. Applied Mathematical Sciences, Vol. 68. Berlin: Springer-Verlag.

Theiler, J. 1990. "Estimating Fractal Dimension." *J. Opt. Soc. Am. A* **7(6)**: 1055–1073.

Theiler, J. 1991. "Some Comments on the Correlation Dimension of $1/f^{\alpha}$ Noise." *Phys. Lett. A* **155**: 480–493.

Theiler, J., P. S. Linsay, and D. M. Rubin. 1993. "Detecting Nonlinearity in Data with Long Coherence Times." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 439-455. Reading, MA: Addison-Wesley.

Tong, H. 1990. *Nonlinear Time Series Analysis: A Dynamical Systems Approach*. Oxford: Oxford University Press.

Tong, H., and K. S. Lim. 1980. "Threshold Autoregression, Limit Cycles and Cyclical Data." *J. Roy. Stat. Soc. B* **42**: 245–292.

Trunk, G. V. 1968. "Representation and Analysis of Signals: Statistical Estimation of Intrinsic Dimensionality and Parameter Identification." *General Systems* **13**:49–76.

Tukey, J. W. 1977. *Exploratory Data Analysis*. Reading, MA: Addison-Wesley.

Turing, A. M. 1936. "On Computable Numbers, with an Application to the Entscheidungsproblem." *Proc. London Math. Soc.* **42**: 230–265.

Ulam, S. 1957. "The Scottish Book: A Collection of Mathematical Problems." Unpublished manuscript. See also the special issue on S. Ulam: *Los Alamos Science* **15** (1987).

Volterra, V. 1959. *Theory of Functionals and of Integral and Integro-Differential Equations*. New York: Dover.

Wallace, C. S., and D. M. Boulton. 1968. "An Information Measure for Classification." *Comp. J.* **11**: 185–195.

Wan, E. A. 1993. "Times Series Prediction Using a Connectionist Network with Internal Delay Lines." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A. S. Weigend and N. A. Gershenfeld, 195–217. Reading, MA: Addison-Wesley.

Weigend, A. S. 1991. "Connectionist Architectures for Time Series Prediction of Dynamic Systems." Ph.D. Thesis, Stanford University.

Weigend, A. S. 1994. "On Overfitting and the Effective Number of Hidden Units." In *Proceedings of the 1993 Connectionist Models Summer School,* edited by M. C. Mozer, P. Smolensky, D. S. Touretzky, J. L. Elman, and A. S. Weigend, 335–342. Hillsdale, NJ: Erlbaum Associates.

Weigend, A. S., and N. A. Gershenfeld, eds. 1993. *Time Series Prediction: Forecasting the Future and Understanding the Past*. Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XV. Reading, MA: Addison-Wesley.

Weigend, A. S., B. A. Huberman, and D. E. Rumelhart. 1990. "Predicting the Future: A Connectionist Approach." *Intl. J. Neur. Sys.* **1**: 193–209.

Weigend, A., B. A. Huberman, and D. E. Rumelhart. 1992. "Predicting Sunspots and Exchange Rates with Connectionist Networks." In *Nonlinear Modeling and Forecasting*, edited by M. Casdagli and S. Eubank. Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XII, 395–432. Redwood City, CA: Addison-Wesley.

Weigend, A. S., and D. E. Rumelhart. 1991a. "The Effective Dimension of the Space of Hidden Units." In *Proceedings of International Joint Conference on Neural Networks, Singapore*, 2069–2074. Piscataway, NY: IEEE Service Center.

Weigend, A. S., and D. E. Rumelhart. 1991b. "Generalization Through Minimal Networks with Application to Forecasting." In *INTERFACE '91—23rd Symposium on the Interface: Computing Science and Statistics*, edited by E. M. Keramidas, 362–370. Conference held in Seattle, WA, in April 1991. Interface Foundation of North America.

Werbos, P. 1974. "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences." Ph.D. Thesis, Harvard University, Cambridge, MA.

White, D. A., and D. A. Sofge, eds. 1992. *Handbook of Intelligent Control*. Van Nostrand Reinhold.

White, H. 1990. "Connectionist Nonparametric Regression: Multilayer Feedforward Networks Can Learn Arbitrary Mappings." *Neur. Net.* **3**: 535–549.

Widrow, B., and M. E. Hoff. 1960. "Adaptive Switching Circuits." In *1960 IRE WESCON Convention Record*, Vol. 4, 96–104. New York: IRE.

Wiener, N. 1949. *The Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications*. New York: Wiley.

Yip, K. M.-K. 1991. "Understanding Complex Dynamics by Visual and Symbolic Reasoning." *Art. Intel.* **51**: 179–221.

Yule, G. 1927. "On a Method of Investigating Periodicity in Disturbed Series with Special Reference to Wolfer's Sunspot Numbers." *Phil. Trans. Roy. Soc. London* **A 226**: 267–298.

Zhang, X., and J. Hutchinson. 1993. "Simple Algorithms on Fast Machines: Practical Issues in Nonlinear Time Series Prediction." In *Time Series Prediction: Forecasting the Future and Understanding the Past*, edited by A. S. Weigend and N. A. Gershenfeld, 219–241. Reading, MA: Addison-Wesley.