20 Linear and Nonlinear Time Series

A Kalman filter, or a Hidden Markov Model, starts with some notion of the dynamics of a system and then seeks to match it to observations. As powerful as these ideas are, what if you're given a signal without *a priori* insight into the system that produced it? What if your goal is to learn more about the nature of the system, not just what its state is? This is the domain of *time series analysis*. The field is as broad as time itself; it is defined not by any particular tools (it draws on many of the preceeding chapters), but rather by the intent of their use.

Time series problems arise in almost all disciplines, ranging from studying variations in currency exchange rates to variations in heart-rates. Wherever they occur there are three recurring tasks:

- *Characterize:* What kind of system produced the signal? How many degrees of freedom does it have? How random is it? Is it linear? How does noise influence the system?
- *Forecast:* Based on an estimate of the current state, what will the system do next?
- *Model:* What are the governing equations for the system? What is their long-term behavior?

These are closely related but not identical. For example, a model with good long-term properties may not be the best way to make short-term forecasts and *vice versa*. And although it's possible to characterize a system without explicitly writing down a model, some of the most powerful characterization techniques are based on first building a model.

This chapter will assume that the analyst is an observer, not a participant. Beyond modeling comes manipulation. If it is possible to influence a system then these kinds of descriptions can be used to choose informative inputs (by selecting them where the model uncertainty is large), and to drive the system to a desired state (by reversing the model to predict inputs based on outputs, the domain of *control theory* [Doyle *et al*., 1992; Auerbach *et al*., 1992]).

Time series originally were analyzed, not surprisingly, in the time domain. Characterization consisted of looking at the series, and the only kind of forecasting or modeling was simple extrapolation. A major step was Yule's 1927 analysis of the sunspot cycle [Yule, 1927]. This was perhaps the first time that a model with internal degrees of freedom (what we would now call a linear autoregressive model) was inferred from measurements of an external observable (the sunspot series). This rapidly bloomed into the theory of linear time series, which is mature, successful, ubiquitous, and applicable only to linear systems. It arises in two very different limits: deterministic systems that are so simple they can be described by linear governing equations, or systems which are so stochastic that their deviation from ideal randomness is governed by linear random variable equations. In between these two extremes lies the rest of the world, for which nonlinearity does matter. The theories of nonlinear differential equations or stochastic processes in general have no general results, but rather there are many particular tractable cases. However, there is a powerful theory emerging for the characterization and modeling of nonlinear systems without making any linear assumptions. This chapter starts with the linear canon and closes with these newer ideas.

20.1 LINEAR TIME SERIES

The most general linear system produces an output y that is a linear function of external inputs x (sometimes called *innovations*) and its previous outputs:

$$y_t = a_t + \underbrace{\sum_{m=1}^{M} b_m y_{t-m}}_{\text{AR, IIR}} + \underbrace{\sum_{n=0}^{N} c_n x_{t-n}}_{\text{MA, FIR}} \quad . \tag{20.1}$$

Typically the a_t term is nonzero only for an initial transient, which imposes the initial conditions on the system. Depending on the side of campus that you are on, the two parts of this equation are called:

- Statistics
  - *Auto-Regressive* (*AR*): The output is a linear regression of its M previous values.
  - *Moving Average* (*MA*): The output is an N-point moving average of the input.
  - Taken together, they define an $ARMA(M,N)$ model.
- Engineering
  - *Infinite Impulse Response* (*IIR*): The output can continue after the input stops.
  - *Finite Impulse Response* (*FIR*): The output stops after the input stops.

For the statistician these are random variables, while the engineer usually tries to make sure that they are not. *Trending* in a nonstationary signal can be removed by differencing it to some order before model building (first-order time differences remove a linear drift, second-order removes a polynomial trend, and so forth), giving an $ARIMA$ (Auto-Regressive Integrated Moving Average) model.

The z-transform of the output

$$Y(z) \equiv \sum_{n=-\infty}^{n=\infty} y_n z^n \tag{20.2}$$

provides a complete analysis of the system (Chapter 3). Since convolution in the time domain equals multiplication in the z domain, the z-transform can easily be solved:

$$y_t = a_t + \sum_{m=1}^{M} b_m y_{t-m} + \sum_{n=0}^{N} c_n x_{t-n}$$

$$\begin{aligned} Y(z) &= A(z) + B(z)Y(z) + C(z)X(z) \\ &= \frac{A(z)}{1-B(z)} + \frac{C(z)}{1-B(z)}X(z) \quad . \end{aligned} \tag{20.3}$$

The z-transform of the output consists of two terms. The first depends on the initial transient, and the second term is equal to the z-transform of the input multiplied by a system *transfer function* that is independent of the input. The output $Y(z)$ consists of a ratio of polynomials $A(z)$ and $C(z)$ divided by $1 - B(z)$ reflecting the system's structure, and a possible non-polynomial part from the input $X(z)$. The numerators, due to the inputs, can have *zeros*, and the denominator, due to the memory of the output, can have *poles*. As we've seen, the location of these poles and zeros determines the system's characteristics (such as stability and oscillation frequencies).

The AR and MA coefficients can be determined from the correlation coefficients. Taking $\langle y \rangle$ to denote the time average expectation value of y (written in the statistics literature as $E[y]$), the *autocorrelation function* is defined to be

$$\begin{aligned} \kappa_\tau &\equiv \frac{\langle (y_t - \langle y_t \rangle)(y_{t-\tau} - \langle y_t \rangle) \rangle}{\langle (y_t - \langle y_t \rangle)^2 \rangle} \\ &= \frac{\langle (y_t - \mu_y)(y_{t-\tau} - \mu_y) \rangle}{\sigma_y^2} \quad . \end{aligned} \tag{20.4}$$

μ_y is the mean and σ_y^2 is the variance. This can also be written as

$$\begin{aligned} \kappa_\tau &= \frac{\langle (y_t - \mu_y)(y_{t-\tau} - \mu_y) \rangle}{\langle (y_t - \mu_y)(y_t - \mu_y) \rangle} \\ &= \frac{\langle y_t y_{t-\tau} \rangle - \mu_y \langle y_{t-\tau} \rangle - \mu_y \langle y_t \rangle + \mu_y \mu_y}{\langle y_t y_t \rangle - \mu_y \langle y_t \rangle - \mu_y \langle y_t \rangle + \mu_y \mu_y} \\ &= \frac{\langle y_t y_{t-\tau} \rangle - \mu_y^2}{\langle y_t y_t \rangle - \mu_y^2} \end{aligned} \tag{20.5}$$

since time averages are independent of the time origin for a stationary process. The autocorrelation function ranges from 1 for perfect correlation between two times, to 0 for uncorrelation, and to -1 for anticorrelation.

For an MA model ($a = b = 0$), if the input is assumed to be zero mean ($\langle x_t \rangle = \mu_x = 0$) then $\mu_y = 0$, and the autocorrelation function becomes

$$\begin{aligned} \kappa_\tau &= \frac{\left\langle \left(\sum_{n=0}^N c_n x_{t-n} \right) \left(\sum_{n'=0}^N c_{n'} x_{t-\tau-n'} \right) \right\rangle}{\left\langle \left(\sum_{n=0}^N c_n x_{t-n} \right) \left(\sum_{n'=0}^N c_{n'} x_{t-n'} \right) \right\rangle} \\ &= \frac{\sum_{n=0}^N \sum_{n'=0}^N c_n c_{n'} \langle x_{t-n} x_{t-\tau-n'} \rangle}{\sum_{n=0}^N \sum_{n'=0}^N c_n c_{n'} \langle x_{t-n} x_{t-n'} \rangle} \quad . \end{aligned} \tag{20.6}$$

If the input x is an uncorrelated stochastic process ($\langle x_i x_j \rangle = 0$ for $i \neq j$) then the MA coefficients are related to the autocorrelation function by

$$\kappa_\tau = \begin{cases} \dfrac{\sum_{n=\tau}^N c_n c_{n-\tau}}{\sum_{n=0}^N c_n^2} & (\tau \leq N) \\ 0 & (\tau > N) \end{cases} \quad . \tag{20.7}$$

Given the c's we can calculate the autocorrelation function, or this relationship can be inverted to find a set of c's to match a given autocorrelation function.

Similarly, multiplying both sides of an AR model by $y_{t-\tau}$ and averaging gives

$$\langle y_t y_{t-\tau} \rangle = \sum_{m=1}^{M} b_m \langle y_{t-m} y_{t-\tau} \rangle \quad , \tag{20.8}$$

and then after normalizing by the variance

$$\kappa_\tau = \sum_{m=1}^{M} b_m \kappa_{\tau-m} \quad . \tag{20.9}$$

Unlike the MA case the autocorrelation function need not vanish after M steps. This linear set of equations, called the *Yule–Walker* equations, can be inverted to relate the AR coefficients to the autocorrelation function. *Levinson–Durbin recursion* is an efficient way to do this [Levinson, 1947; Durbin, 1960].

Unlike the simplicity of AR and MA models there is not a unique algorithm to find the best ARMA model to describe a data set (but there is a lot of heated debate about how to do it). The *Box–Jenkins* procedure is a popular recursive solution [Box *et al.*, 1994]. It's always possible to trade off more or less of M versus N in selecting the order of an ARMA model; the *Akaike Information Criteria* (*AIC*) and its Bayesian cousin the *BIC* do this by assigning an informational cost to the number of parameters, to be minimized along with the model error [Akaike, 1979].

20.2 THE BREAKDOWN OF LINEAR SYSTEMS THEORY

The essence of linear systems theory is expressed by the *Wold Decomposition*: any stochastic process can be separated into the sum of two processes – a deterministic one that is a linear function of its past values, and a stochastic one that is a linear function of previous values of an uncorrelated random variable [Priestley, 1981]. Once these two pieces have been found there is nothing more that can be said about the system.

If you limit yourself to linear models, that is. Even simple nonlinearities can be completely misunderstood by a linear analysis. Consider the two simple iterated maps shown in Figure 20.1. The first one,

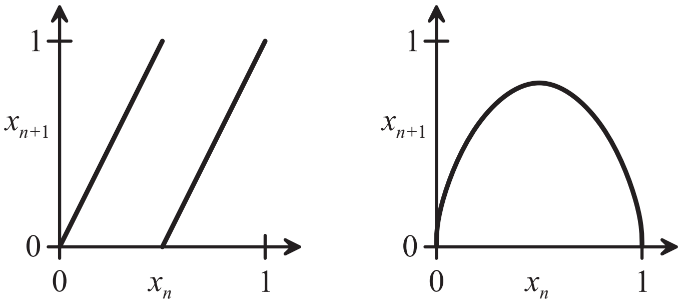
$$x_{n+1} = 2x_n \text{ (mod 1)} \tag{20.10}$$

is called the *mod map* and it shifts every bit of x (written in a binary fractional expansion) over one place and then discards the most significant bit. This means that the trajectory of the system (for example, which branch it is on) is determined solely by the initial condition. If the initial condition is a real number with digits that appear to be random, say π, then this map will generate a broadband power spectrum. There is a simple truth (equation 20.10), but a linear model is forced to find the best single straight line to fit what is really two straight lines. The mismatch can only be attributed to stochastic inputs.

The second map

$$x_{n+1} = \lambda x_n (1 - x_n) \tag{20.11}$$

(the logistic map) arises in a variety of systems such as chemical reactions, electrical circuits, hydrodynamic flows, and population dynamics, because of the universal properties



$x_{n+1} = 2\, x_n \pmod 1$ $x_{n+1} = \lambda\, x_n\, (1\text{-}x_n)$

Figure 20.1. Two simple nonlinear maps.

of a smooth map with a peak (the behavior is determined by the lowest-order term in a Taylor expansion around a maximum) [Collet & Eckmann, 1980]. For $\lambda = 4$ each iteration of this simple chaotic map once again reveals one bit of information from the initial condition, and once again the output cannot be described with linear systems theory.

The problem in both cases is that the mod map and the logistic map are poorly approximated by a straight line (a linear model); the same is true in higher dimensions, where a linear model is equivalent to a hyperplane. These examples suggest two generalizations [Priestley, 1991]:

- *TAR* (Threshold Autoregressive): partition the space into two or more regions, each with a different linear model.
- *Volterra series*, *NAR* (Nonlinear Autoregressive): include bilinear and higher-order terms in the model.

The recognition of these alternatives is almost as old as Yule's original analysis, but the problem with them is that some extra kind of insight is needed to decide how to partition the space or chose the higher-order terms. Our first step in addressing these questions will be to bring in a deeper notion of the state of a nonlinear system.

20.3 STATE-SPACE RECONSTRUCTION

Embedding or *state-space reconstruction* is best introduced with a simple example. Consider the *Lorenz set* of three coupled nonlinear first-order differential equations

$$
\begin{aligned}
\dot{x} &= \sigma(y - x) \\
\dot{y} &= -xz + bx - y \\
\dot{z} &= xy - rz
\end{aligned}
$$

$$\sigma = 10,\ b = 8/3,\ r = 28 \quad , \tag{20.12}$$

studied by Ed Lorenz at MIT as the first terms in a Galerkin approximation of atmospheric convection [Lorenz, 1963]. Although they retain little connection to the original atmospheric problem, they do exactly describe convection in a thin annular ring heated

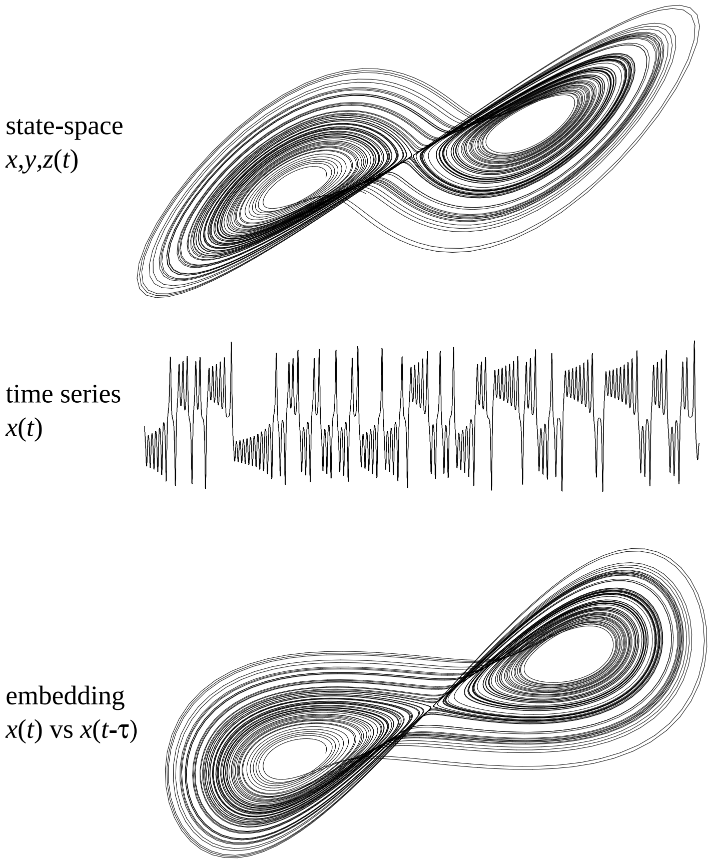


Figure 20.2. Embedding the Lorenz set.

asymmetrically, where the variables are the temperature and pressure gradient and the angular velocity. The top part of Figure 20.2 shows the trajectory in the x, y, z space, looking down on the z axis. The time series of $x(t)$ is shown in the middle, discarding any knowledge of y, z, or the governing equations. This looks rather random, and indeed has a broadband power spectrum. The bottom part of the figure takes the time series and plots $x(t)$ versus $x(t - \tau)$ (where τ is a fixed arbitrary delay). It looks very similar to the upper plot of x, y, and z: even though it is stretched, all of the details are the same. The theory of embedding explains why this is not just a remarkable coincidence but is a deep property of time-lag spaces.

It is important to be clear about the spaces relevant to embedding, shown in Figure 20.3:

(a) *The physical degrees of freedom*: For example, a convecting fluid is described by the continuous (infinite-dimensional) fields of the Navier–Stokes PDEs.

(b) *The configuration state-space*: In this space a continuous time ($\dot{\vec{x}} = \vec{f}(\vec{x})$) or discrete time ($\vec{x}_{n+1} = \vec{f}(\vec{x}_n)$) set of equations governs the evolution of the state vector $\vec{x}$, which describes a distinguishable macroscopic state of the system (such as the temperature, pressure, and angular variables in the Lorenz set). Dissipation and symmetry will reduce a PDE to such a set of coupled ODEs [Temam, 1988].

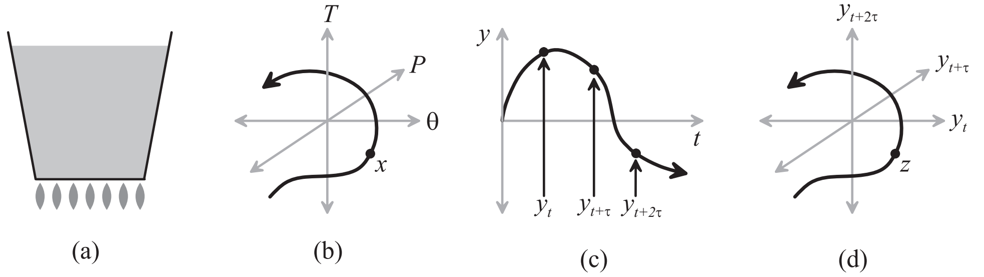


Figure 20.3. The spaces relevant to embedding.

(c) *The experimental observable*: This is an accessible continuous time $y(\vec{x}(t))$ or discrete time $y(\vec{x}_n)$ observable (or set of observables) that depends on the internal configuration variables, such as a temperature or velocity probe in a convecting fluid.

(d) *The reconstructed state-space*: This space is most simply found by time lags $\vec{z}(t) = (y_t, y_{t+\tau}, \ldots, y_{t+(D-1)\tau})$. According to the embedding theorem, the trajectory in this lag space produced from a scalar observable differs from the trajectory in the original configuration space by no more than a smooth local change of coordinates.

Unless specifically noted, all the results in this chapter apply equally to continuous time flows and discrete time maps.

Embedding is both a simple well-known prescription (use time lags) and a profound insight (an accessible variable can explicitly retrieve unseen internal degrees of freedom). Time lags have long been used in studying signals; what was new was the recognition that there is a deep connection between them and a system's inaccessible internal state. This was first proved by Floris Takens [Takens, 1981], and the idea was suggested by David Ruelle, and others [Packard *et al*., 1980]. In the original form it states that the configuration space vector $\vec{x}(t)$ and the one that is reconstructed from a time series $\vec{z}(t) = (y_t, y_{t+\tau}, \ldots, y_{t+(d-1)\tau})$ will *generically* differ by no more than a smooth invertible local change of coordinates (i.e., it is an *embedding*, which is locally *diffeomorphic*) for all τ as long as d is large enough, y depends on at least some of the components of $\vec{x}$, and the other components are coupled by the dynamics. If the system's trajectory is thought of as being printed on a rubber sheet, according to the diffeomorphic property it can be stretched but not cut. The word embedding is used in the literature both in this exact technical topological sense, and more loosely to refer to the entire procedure.

The embedding theorem is true "generically." This means that there are isolated cases for which it will fail (such as sampling a sine wave exactly at its period) but that these will be removed by a small perturbation (such as sampling a sine wave at the period plus a small ϵ). In practice, the problems that embedding encounters come not from this assumption of genericity but from real-world problems such as short nonstationary data sets. Although the embedding theorem holds "for all" values of the time lag τ, in practice this will not be the case. As τ increases from 0 the embedding grows away from the diagonal of the embedding space (all the lags are equal), but for small τ this can be masked by noise or a finite sampling resolution. If τ becomes too large, it can bring distant parts of the trajectory accidentally close together, which can once again be masked by noise or the sampling resolution. The last qualifier, "large enough", is explained by the *Whitney embedding theorem*: an arbitrary D-dimensional manifold can always be embedded in a $2D$-dimensional Euclidean space (although D may be sufficient) [Guillemin & Pollack, 1974]. For example, a sheet of paper (which is two-dimensional) can be embedded into a 2D space. If the ends of the sheet of paper are joined to form a loop (or with a twist, forming a Möbius strip), then it will require a 3D embedding space. If the opposite ends are joined, forming a Klein bottle, then a 4D space is needed to avoid crossings. That's the maximum; according to the Whitney theorem, no more than 4D can be needed. We will return to these qualifiers in the Section 20.4.

The proof of the embedding theorem has two parts: one showing that local properties are preserved, such as the rate of divergence of trajectories, and the other showing that global properties are preserved, such as the linking of trajectories. The governing equations $\dot{\vec{x}} = \vec{f}(\vec{x})$ imply a solution function $\vec{x}_{t+\tau} = \vec{\varphi}_\tau(\vec{x}_t)$, which for any nontrivial system is hopelessly complicated and cannot be written down in a closed form. This (unknown) solution function in turn implies an embedding mapping function

$$\vec{z}_t \;=\; \vec{\Phi}(\vec{x}_t) \;=\; y(\vec{x}_t), y(\vec{\varphi}_\tau(\vec{x}_t)), \ldots, y(\vec{\varphi}_{t+(d-1)\tau}(\vec{x}_t)) \quad . \tag{20.13}$$

The local behavior of a point is given by the linearization of this map

$$\vec{z} + d\vec{z} = \vec{\Phi}(\vec{x}) + (D\vec{\Phi}) \cdot d\vec{x} \quad . \tag{20.14}$$

The local part of the embedding proof follows if this map is of full rank (basis vectors in the $\vec{x}$ space span the $\vec{z}$ space under the mapping). A typical term of the derivative of the map is

$$(D\vec{\Phi})_{ij} = \frac{\partial \Phi_i}{\partial x_j} = \frac{\partial y(\vec{\varphi}_{i\tau}(\vec{x}))}{\partial x_j} \quad ; \tag{20.15}$$

for the mapping not to be of full rank, one column of this matrix must be proportional to another. Since this mixes up shifts in time with shifts among variables it requires a tremendous degeneracy to occur, which will be removed by almost any small perturbation to the problem. Making this plausible argument rigorous follows from the theory of *transversality*. An example of a transversality result is the observation that the intersection between two ropes will usually remain if the ropes are moved in 2D, but not in 3D. The global part of the embedding proof follows from the uniqueness of solutions to differential equations.

The embedding result that time lags retrieve internal degrees of freedom appears similar to traditional engineering practice, which is full of the use of time lags and "state-space" models. The three key new insights are that time lags are not just convenient, they reconstruct all of the internal degrees of freedom that influence the observable; once enough time lags are used on a noise-free system (within a factor of 2 of the degrees of freedom of the system, depending on the complexity of the geometry) then there is nothing more to be learned by more lags; and even though the embedding space has been stretched by an unknown change of coordinates it is still possible to characterize, predict, and model using topologically invariant quantities.

Embedding has since been generalized in a few directions. One important result is that any linear transformation of a time-lagged vector is an embedding, with the embedding dimension being given by the rank of the (not necessarily square) transformation matrix [Sauer *et al.*, 1991]. This means that embedding can be related to signal processing. For example, it is possible to do noise reduction along with embedding by taking an FFT, applying a filter, and then taking an inverse FFT, by using a wavelet transform and retaining some coefficients, or applying a circulant filter matrix. A related result is that if a time series consists of distinguishable events (such as pulses), the times between the events can be used for embedding.

A second generalization of embedding comes from recognizing that the probability distribution $p(\vec{z})$ in the embedding space, needed for most characterization and prediction algorithms, is defined in terms of an arbitrary test function $f(\vec{z})$ by

$$\langle f(\vec{z}_t)\rangle_t = \lim_{T\to\infty}\frac{1}{T}\int_0^T f(\vec{z}_t)dt = \int f(\vec{z})\; p(\vec{z})\; d\vec{z} \quad . \tag{20.16}$$

Note that this is not the same as assuming that the signal is *ergodic* (in an ergodic system all trajectories have the same long-term time average, which can be written as a space average); it is simply the definition of a probability distribution for an observed signal. If a complex exponential is used for the test function

$$\langle \mathrm{e}^{i\vec{k}\cdot\vec{z}_t}\rangle_t = \langle \mathrm{e}^{i\vec{k}\cdot(y_t,y_{t+\tau},\ldots,y_{t+(d-1)\tau})}\rangle_t = \int \mathrm{e}^{i\vec{k}\cdot\vec{z}}p(\vec{z})\; d\vec{z} \tag{20.17}$$

we see that the time average is the Fourier transform of the embedded probability distribution, permitting embedding to be done without recording time series if the expectations can be directly measured, and providing a way to separate an unknown signal from known noise (a nonlinear generalization of a Wiener filter). This is a time-lagged *characteristic function*; its power series expansion is

$$\begin{aligned}\langle \mathrm{e}^{i\vec{k}\cdot\vec{z}_t}\rangle_t &= \int \mathrm{e}^{i\vec{k}\cdot\vec{z}}p(\vec{z})\; d\vec{z} \qquad (20.18)\\ &= \sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty}\cdots\sum_{n_d=0}^{\infty}\\ &\quad \frac{(ik_1)^{n_1}}{n_1!}\frac{(ik_2)^{n_2}}{n_2!}\cdots\frac{(ik_d)^{n_d}}{n_d!}\langle y_t^{n_1}y_{t+\tau}^{n_2}\cdots y_{t+(D-1)\tau}^{n_d}\rangle_t \quad .\end{aligned}$$

This shows that the probability distribution depends on a infinite family of powers of multiple-time correlation functions; the measurements used in linear systems theory are based on just the first few terms of this series (single-time correlations for the mean and variance, two-time correlations for the spectrum, and three-time correlations for the *bispectrum*). It also provides a connection between embedding and deterministic expectation values of functions of the random variables of a stochastic process.

So far we have been discussing embedding *autonomous* systems that have no external inputs, but most of the world is not so obliging as to stay still while we watch it. A much more relevant case is

$$\frac{d\vec{x}}{dt} = \vec{f}(\vec{x},\vec{u}) \quad , \tag{20.19}$$

where $\vec{u}$ is a vector of the inputs to a system. If the inputs are known then embedding

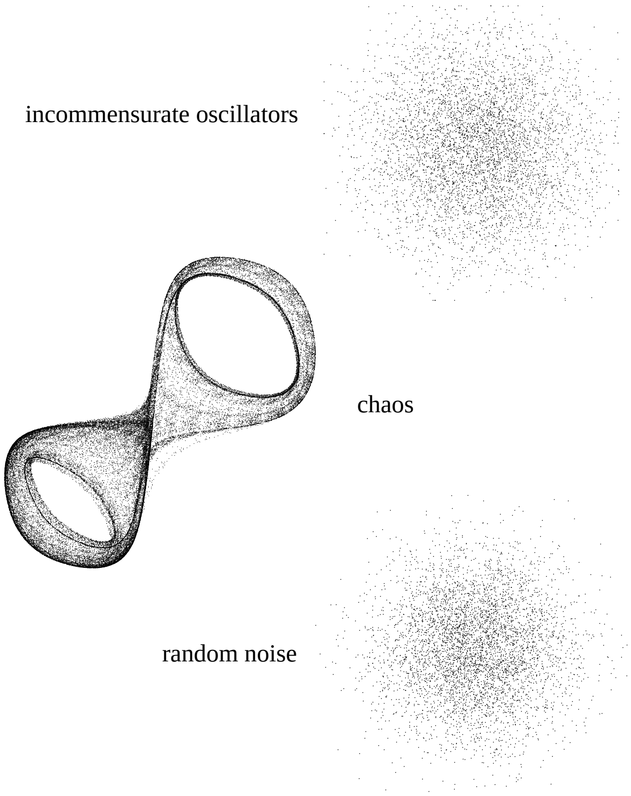


Figure 20.4. The need for characterization.

using lags of both the inputs and the outputs will still recover the internal state [Casdagli, 1992; Stark *et al*., 1997].

20.4 CHARACTERIZATION

We could look at Figure 20.2 and instantly recognize the simple structure, but this will not usually be the case. Figure 20.4 shows the 2D embedding for three electrical systems [Gershenfeld, 1992]. The bottom one is Johnson noise, the thermodynamic fluctuations in an electrical resistor. This is a random Gaussian stochastic process, and the embedding is a Gaussian spot. The middle system is a chaotic nonlinear circuit, and the low-dimensional dynamics is readily apparent. However, the top example is a 2D embedding of the sum of 12 incommensurate electrical oscillators. The 2D projection of a 12D torus looks Gaussian, and in fact it approximately is (by the Central Limit Theorem). Some kind of characterization is needed to "see" in 12D to recognize that this is just a torus.

The simplest kinds of characterization are basic sanity checks, easy to forget amid advanced algorithms. To start, is the time series stationary? Are its statistical properties the same at the beginning and the end? If not, a naive analysis will mix unrelated behaviors of the system. Are there obvious features that a model must include (such as non-negativity)? It's important to look at data to recognize these basic features.

Linear correlations in a data set can mask nonlinear structure. For a 2D embedding, the moments of the probability distribution are

$$\begin{aligned} M_{ij} &= \int \int x_i x_j p(x_i, x_j) \; dx_i dx_j \\ &= \lim_{T\to\infty} \frac{1}{T} \int_0^T x_i(t) x_j(t) \; dt \quad . \end{aligned} \tag{20.20}$$

Therefore, $M_{11} = M_{22} =$ the variance σ^2, and $M_{12} = M_{21} = \sigma^2\kappa(\tau)$ (where $\kappa(\tau)$ is the autocorrelation function). The ratio of the eigenvalues of this correlation matrix gives the width-to-length ratio of the embedded probability distribution

$$\frac{\lambda_-}{\lambda_+} = \frac{\kappa(0) - \kappa(\tau)}{\kappa(0) + \kappa(\tau)} \quad . \tag{20.21}$$

For a power-law power spectrum $S(\omega) = |\omega|^{-\alpha}$ ($\alpha = 1 =$ *1/f noise*, $\alpha = 2 =$ *diffusion noise*, ...), with a bandwidth of 10^{-3}–10^3 Hz and a delay $\tau = 1$ s, $\alpha = 1 \Rightarrow \lambda_-/\lambda_+ = 0.51$, $\alpha = 2 \Rightarrow \lambda_-/\lambda_+ = 0.005$, $\alpha = 3 \Rightarrow \lambda_-/\lambda_+ = 0.0001$. As α is increased the distribution becomes so skinny that given finite sampling resolution it will erroneously appear to be one-dimensional at a fixed τ [Gershenfeld, 1992].

An important sanity check for linear structure in a presumed nonlinear signal is to use *surrogate data* [Theiler *et al.*, 1992]. In the simplest form, these are generated by taking the Fourier transform of the real-valued time series $y(t)$ to find $Y(\omega) = A(\omega)e^{i\theta(\omega)}$, then setting the phases $\theta(\omega)$ to random values symmetrically ($Y(-\omega) = Y(\omega)$), and transforming back. The resulting series will be real-valued, and because $A(\omega)$ is unchanged it will have the same power spectrum (and autocorrelation function), but any nonlinear relationship among the points will have been randomized. If the result of a test is the same on the original and the surrogate data, the test can only be sensitive to the linear structure in the data. More sophisticated versions fit an ARMA model and generate new realizations of the stochastic process.

20.4.1 Dimensions

Dimension measurement was one of the first widespread characterization techniques used in embedding spaces [Grassberger & Procaccia, 1983a]. Although it is now used less commonly in favor of more powerful and reliable techniques, it is an important concept for describing a set of points. Define a point correlation function for a data set of N points by the number of neighbors in a hypersphere of radius r around a reference point

$$C_i(r) \equiv \frac{1}{N} \text{ (the number of } \vec{z}_j \text{ within } r \text{ of a reference point } \vec{z}_i) \quad , \tag{20.22}$$

and average this to get the *radial correlation function*

$$C(r) \equiv \frac{1}{N} \sum_{i=1}^{N} C_i(r) \quad . \tag{20.23}$$

In general, in the limit of small r this will scale as a power of the distance (the first power for points distributed along a line, the second power for points on a surface, ...), and so this exponent defines the *correlation dimension* ν

$$C(r) \approx Ar^{\nu} \quad . \tag{20.24}$$

Taking the logarithm,

$$\log C(r) = \log A + \nu \log r$$

$$\frac{\log C(r)}{\log r} = \frac{\log A}{\log r} + \nu \quad . \tag{20.25}$$

In the limit $r \to 0$ the first term on the right hand side will vanish, and so

$$\nu \equiv \lim_{r \to 0} \frac{\log C(r)}{\log r} \quad . \tag{20.26}$$

If ν is not an integer, the data set is a *fractal*.

The correlation dimension is invariant under a smooth change of coordinates (the embedding). In practice, the correlation dimension is measured for successively larger embedding dimensions. If the data are random they will fill the embedding space and the measured dimensions will equal the embedding dimension, but if the data are deterministic then the measured dimension will stop growing once the embedding dimension is reached. At this point the measured dimension (or the smallest integer greater than the measured dimension) gives the number of local degrees of freedom needed to describe a state of the system.

The number of data points needed to estimate a dimension can roughly be estimated by recognizing that $C(r) \sim r^{\nu}$, therefore if a reliable estimate of the slope on a log–log plot requires a decade of scaling ($r_0 \to 10\ r_0$), then the number of points required is 10^{ν} (with some controversy over the base of the exponent). This simple argument is not really correct, because the real problem is that in a high-dimensional space all of the points are near the surface of the distribution, and so it becomes impossible to record a "typical" point in the interior (Section 14.5). This results in a data requirement for small dimensions that is weaker than that predicted by a simple exponential estimate, but for large dimensions it is much worse. For sane amounts of data (say, 10^7 points) from an ideal stationary system, this limits measured dimensions to be less than roughly 10–20 degrees of freedom.

Correlation dimensions can be generalized to an infinite family of *generalized dimensions* based on the scaling of moments of the probability distribution, estimated by covering with boxes of side length l:

$$\begin{aligned}
D_q &\equiv \lim_{l \to 0} \frac{1}{q-1} \frac{\log_2 \sum_i p_d(\vec{x}_i)^q}{\log_2 l} \\
D_0 &\equiv \lim_{l \to 0} -\frac{\log_2 \sum_i p_d(\vec{x}_i)}{\log_2 l} \quad \text{(Hausdorff dimension)} \\
D_1 &\equiv \lim_{l \to 0} \frac{\sum_i p_d(\vec{x}_i) \log_2 p_d(\vec{x}_i)}{\log_2 l} \quad \text{(Information dimension)} \\
D_2 &\equiv \lim_{l \to 0} \frac{\log_2 \sum_i p_d(\vec{x}_i)^2}{\log_2 l} \quad \text{(Correlation dimension)} \quad .
\end{aligned} \tag{20.27}$$

The information dimension follows from taking the limit, and the second moment is equal to the correlation dimension because the square of the number of points in a box is equal to the number of pairs of points in the box [Gershenfeld, 1989]. The D_q are equal for *homogeneous fractals*, and their spread measures how singular the distribution is. Their values are closely related [Beck, 1990]

$$D_2 \geq D_\infty \geq D_2/2 \geq 0 \quad , \tag{20.28}$$

therefore any of these can be used to test for degrees of freedom.

20.4.2 Lyapunov Exponents

Dimension measurement provides a way to estimate the number of degrees of freedom of a system without requiring an explicit model of the system, but it provides no insight into the time evolution of the system. *Lyapunov exponents* are one common way to characterize the time dependence.

If

$$\frac{d\vec{x}}{dt} = \vec{f}(\vec{x}) \quad , \tag{20.29}$$

then a small displacement around a point

$$\frac{d\vec{x} + \vec{\delta}}{dt} = \vec{f}(\vec{x} + \vec{\delta}) \approx \vec{f}(\vec{x}) + (D\vec{f}) \cdot \vec{\delta} \tag{20.30}$$

will evolve according to the linear differential equation

$$\frac{d\vec{\delta}}{dt} = (D\vec{f}) \cdot \vec{\delta} \quad . \tag{20.31}$$

The Lyapunov exponents λ_i are the time average of the eigenvalues of the local linearization $(D\vec{f})$.

Lyapunov exponents are preserved under a locally linear change of coordinates and thus can be measured on embedded data. Locally, positive exponents correspond to directions that expand exponentially, and negative exponents to directions that contract exponentially. If an ensemble of trajectories are started in a small hypersphere, the exponents give the growth or contraction rates $\exp(\lambda_i t)$ of the principal axes of the resulting ellipsoid. The sum of all the exponents is the volume expansion rate (zero for conservative systems, negative for dissipative systems), and the sum of the positive exponents is the rate at which new information enters the system (we will revisit this in the next section). A chaotic system has one or more positive exponents but a negative total sum, and a Hamiltonian stochastic system has one or more positive exponents and a total sum of zero. The exponents are related to the dimension by the Kaplan–Yorke conjecture [Frederickson *et al.*, 1983]

$$D_1 = j + \frac{\sum_{i=1}^{j} \lambda_i}{|\lambda_{j+1}|} \qquad \left(\sum_{i=1}^{j} \lambda_i > 0 \ , \quad \sum_{i=1}^{j+1} \lambda_i < 0 \right) \tag{20.32}$$

(ordering the exponents such that $\lambda_1 > \lambda_2 > \ldots$). This is an explicit statement of the connection between dissipation and dimensional reduction, underlying the applicability of embedding to systems governed by PDEs.

Dissipation in chaotic systems requires state-space volume to contract, positive exponents lead to continuous divergence of trajectories, but trajectories cannot cross because of uniqueness of solutions. The only way to satisfy these conflicting requirements is for the trajectories to lie on an infinitely interleaved *strange attractor*.

20.4.3 Entropies

*** cf random variables chapter ***

Correlation functions, the essential tool for measuring dependencies in a linear system, are useless for nonlinear systems. Signals from even simple nonlinear systems can have broadband power spectra and hence featureless correlation structure. Information-theoretic quantities provide an elegant alternative that captures the essential features of a correlation function, and more. Entropy has a long history in dynamics starting with Boltzmann and kinetic theory, and evolving through Szilard's one-atom analysis of Maxwell's demon, to Shannon's information theory, and more recently ergodic theory due to Kolmogorov and others [Leff & Rex, 1990]. More recently, Shaw and Fraser have helped reintroduce entropy back to its roots in dynamics [Shaw, 1981; Fraser, 1989; Gershenfeld, 1993].

Let's start by assuming that we have an observable quantized to one of N integer values

$$y(t) \in \{1, \ldots, N\} \tag{20.33}$$

(we will soon examine the dependence on N). From measurements of y we can estimate the probability distribution; naively this can be done by binning, taking the ratio of the number of times a value was seen to the total number of points

$$p_1(y) = n_y / n_T \quad . \tag{20.34}$$

Chapter 16 discusses the limitations of, and alternatives to, this simple estimator.

The *entropy* of this distribution is given by

$$H_1(N) = -\sum_{y=1}^{N} p_1(y) \log_2 p_1(y) \quad . \tag{20.35}$$

It is the average number of bits required to describe a sample taken from the distribution, i.e., the expected value of the *information* in a sample. The entropy is a maximum if the distribution is flat (we don't know anything about the next point), and a minimum if the distribution is sharp (we know everything about the next point). Similarly, for a point $\vec{z}$ in the lag space we can ask for the joint probability to see the corresponding sequence in the time series

$$p_d(y_t, y_{t-\tau}, \ldots, y_{t-(d-1)\tau}) \approx n_{\vec{z}} / n_T \quad , \tag{20.36}$$

and measure the *block entropy*

$$H_d(\tau, N) = -\sum_{y_t=1}^{N} \cdots \sum_{y_{t-(d-1)\tau}=1}^{N} p_d \log_2 p_d \tag{20.37}$$

which gives the average number of bits needed to describe the sequence. In the limit of lag time going to zero, all of the points become the same and so the block entropy becomes equal to the scalar entropy

$$\begin{aligned}\tau \to 0 &\Rightarrow p_d(y_t, y_{t-\tau}, \ldots, y_{t-(d-1)\tau}) = p_1(y) \\ &\Rightarrow H_d(0, N) = H_1(N) \quad . \end{aligned} \tag{20.38}$$

On the other hand, in the limit of long time lags, if the points become independent (the probability distribution factors), then the block entropy becomes d times the scalar entropy

$$\lim_{\tau\to\infty} p_d(y_t, \ldots, y_{t-(d-1)\tau}) = p_1(y_t)p_1(y_{t-\tau}), \ldots, p_1(y_{t-(d-1)\tau}) \tag{20.39}$$

$$\Rightarrow H_d(\tau, N) = dH_1(N)$$

The connection between entropy and dimensions comes from recognizing that [Gershenfeld, 1989]

$$\lim_{q\to 1} D_q = \lim_{N\to\infty} \frac{\sum_{\vec{z}_i} p_d(\vec{z}_i) \log_2 p_d(\vec{z}_i)}{\log_2 N} = \lim_{N\to\infty} \frac{H_d(\tau, N)}{\log_2 N} \quad . \tag{20.40}$$

The scaling of the block entropy with resolution is just the (information) dimension, giving the number of local degrees of freedom of the system.

The *mutual information* is defined to be the difference in the information between two samples taken independently and taken together

$$\begin{aligned} I_2(\tau, N) &= -\sum_{y_t=1}^{N} p_1(y_t) \log_2 p_1(y_t) \\ &\quad - \sum_{y_{t-\tau}=1}^{N} p_1(y_{t-\tau}) \log_2 p_1(y_{t-\tau}) \\ &\quad + \sum_{y_t=1}^{N} \sum_{y_{t-\tau}=1}^{N} p_2(y_t, y_{t-\tau}) \log_2 p_2(y_t, y_{t-\tau}) \\ &= 2H_1(\tau, N) - H_2(\tau, N) \quad . \end{aligned} \tag{20.41}$$

If the points don't depend on each other then the mutual information is zero:

$$p_2(y_t, y_{t-\tau}) = p_1(y_t)p_1(y_{t-\tau}) \;\Rightarrow\; I_2(\tau, N) = 0 \quad , \tag{20.42}$$

and if they are completely dependent then the mutual information is equal to all of the bits in one sample:

$$p_2(y_t, y_{t-\tau}) = p_1(y_t) \;\Rightarrow\; I_2(\tau, N) = H_1 \quad . \tag{20.43}$$

Here then is an alternative to correlation functions, measuring the connection between two variables without assuming any functional form other than what is needed to estimate a probability distribution.

The *redundancy* extends mutual information to higher-dimensional spaces [Fraser, 1989]. It is equal to the information in one sample, plus the previous $d - 1$ samples, minus the information in d samples:

$$R_d(\tau, N) = H_1(\tau, N) + H_{d-1}(\tau, N) - H_d(\tau, N) \quad . \tag{20.44}$$

If the new point doesn't depend on the previous points,

$$p_d(y_t, \ldots, y_{t-(d-1)\tau}) = p_1(y_t) p_{d-1}(y_{t-\tau}, \ldots, y_{t-(d-1)\tau})$$
$$\Rightarrow H_d = H_1 + H_{d-1} \Rightarrow R_d = 0 \quad , \tag{20.45}$$

then the redundancy is zero, and if the point is completely determined by the previous points,

$$p_d(y_t, \ldots, y_{t-(d-1)\tau}) = p_{d-1}(y_{t-\tau}, \ldots, y_{t-(d-1)\tau})$$
$$\Rightarrow H_d = H_{d-1} \Rightarrow R_d = H_1 \quad , \tag{20.46}$$

then the redundancy is equal to the scalar entropy (all of the bits in the point). The redundancy measures how necessary the bits of the new point were given the past history.

The asymptotic growth rate of the block entropy is called the *source entropy*

$$h(\tau, N) = \lim_{N\to\infty} \lim_{d\to\infty} H_d(\tau, N) - H_{d-1}(\tau, N) \quad . \tag{20.47}$$

The limits don't need to go to infinity. The limit in d is reached at the embedding dimension (if there is one), and the limit in N is reached if there is a *generating partition* (which means that the intersection of all future and backwards iterates of the grid of points at the maximum resolution can be used to specify any neighborhood to an arbitrary precision). Since

$$\tau = 0 \Rightarrow H_{d-1}(0, N) = H_d(0, N) \Rightarrow R_d(0, N) = H_1(N) \quad , \tag{20.48}$$

we see that for small time lags the decay of the redundancy gives the source entropy

$$R_d(\tau, N) = H_1(N) - \tau h(1) \quad . \tag{20.49}$$

It should not be surprising that the source entropy, which is the rate at which information enters the system, is proportional to the volume divergence rate (the sum of positive exponents)

$$h(\tau) = \tau h(1) = \tau \sum_i \lambda_i^+ \quad . \tag{20.50}$$

This is called *Pesin's identity* [Liu & Qian, 1995].

Summarizing: for small time lags, the growth rate of the redundancy with resolution gives the effective sample resolution. As τ is increased, if d is below the embedding dimension then the redundancy will drop quickly to zero. Once d exceeds the embedding dimension (if there is one), the redundancy at small lags will jump up to the scalar entropy. The slope as τ is increased then gives the source entropy (the sum of positive exponents), and the growth rate of the entropy with resolution gives the information dimension (local degrees of freedom). The difference between the embedding dimension and the information dimension will depend on how complex the geometry is. Finally, the time lag at which the redundancy falls to zero gives the predictability horizon at that resolution, beyond which the system's state depends on information that you don't have access to. If a time series is analyzed in reverse order, positive and negative exponents switch, allowing the sum of the negative exponents to be estimated. Since flows (continuous

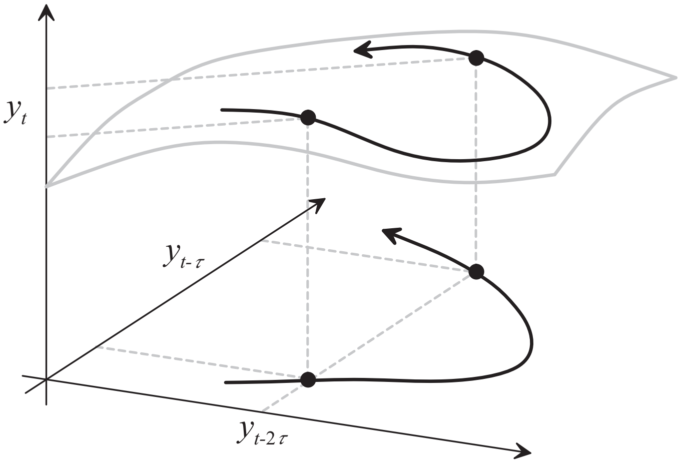


Figure 20.5. Forecasting in a lag space.

time dynamics) have a unique inverse but maps may not, the difference in forward and backward redundancy is a test of the invertibility of the dynamics.

Whew – that's a lot of information to squeeze out of a single statistic. Entropy measurements provides insight into many essential aspects of a system without imposing any specific model in advance; the results of this analysis have significant implications for how a system should best be described and understood. The only catch is that it's based on estimating entropy, which as we saw in Section 16.1 is a tricky thing to do reliably. The problem is that there's no notion of generalization built into a simple entropy estimate. Because of this it can require an enormous amount of data, and be difficult to falsify by making predictions. This is why characterization is intimately connected with forecasting, to which we turn next.

20.5 FORECASTING

One of the most important properties that is preserved under a smooth change of coordinates is trajectory crossing (or lack thereof): if a system's internal state is a deterministic function of its past, then the same will hold in an embedding space. As shown in Figure 20.5, this means that for a deterministic system once enough lags are used the dynamics will lie on a single-valued surface. Forecasting in this case reduces to modeling the shape of the surface. This is exactly the kind of fitting problem covered previously, and presents the same kinds of choices [Weigend & Gershenfeld, 1993].

One approach is to seek a global representation for the prediction function $y_{t+1} = f(\vec{z}_t) = f(y_t, y_{t-\tau}, ..., y_{t-(d-1)\tau})$. A convenient way to do this is to use the functional orthogonalization covered in Section 14.2 [Giona *et al.*, 1991]. Let $p(\vec{z})$ be the probability distribution in the lag space, and $\{f_i(\vec{z})\}$ be a family of functions such as polynomials that have been constructed by Gram–Schmidt orthogonalization to be orthonormal with respect to it

$$\langle f_i(\vec{z}) f_j(\vec{z}) \rangle_t = \int f_i(\vec{z}) f_j(\vec{z}) p(\vec{z})\, d\vec{z} = \delta_{ij} \quad . \tag{20.51}$$

The expansion coefficients that we want are

$$f(\vec{z}) = \sum_i a_i f_i(\vec{z}) \quad . \tag{20.52}$$

By construction, these can be found from the orthonormality condition by summing over the data:

$$a_i = \langle f(\vec{z}_t) f_i(\vec{z}_t)\rangle = \langle y_{t+1} f_i(\vec{z}_t)\rangle = \frac{1}{N}\sum_{t=1}^{N} y_{t+1} f_i(\vec{z}_t) \quad . \tag{20.53}$$

This is certainly easy to implement, and can work well if the surface is not too complicated, but like all global functions it has trouble capturing local features. An alternative is to replace the global predictor by a family of local models [Farmer & Sidorowich, 1987]; this requires some strategy for deciding where to place the models and how to size their neighborhoods (more about that in a moment).

So far we've been discussing *point predictions* where we try to determine future values. This makes sense if the system is nearly deterministic but is pointless if it is effectively stochastic. Then, instead of trying to forecast a random variable, it's necessary to model the expected value of observables such as the power spectral density, or the variance of a process (which is all that's needed to get rich on Wall Street [Hull, 2008]). It's possible to forecast a variance by training a second model to predict the errors of a point prediction model; better still is to do some kind of density estimation to be able to answer other questions as well. And given a model of the noise it's possible to do more than just describe it; self-consistently separating it can reduce the noise in a signal, providing a nonlinear analog to the Wold decomposition [Abarbanel *et al.*, 1993; Weigend *et al.*, 1996].

These issues of modeling stochasticity and introducing locality can be addressed by cluster-weighted modeling, using kernel density estimation with local models (Section 16.4). Figure 20.6 shows part of an example time series, a laser fluctuating near the gain threshold (data set A from [Weigend & Gershenfeld, 1993]), and Figure 20.7 shows the resulting model using linear covariance clusters.

Cluster parameters in a lag space can be related to the time series techniques described earlier. For example, in terms of the eigenvalues of the cluster-weighted covariance matrix $\mathbf{C}_m = \{\sigma_{1,m}, \ldots, \sigma_{D,m}\}$, the radial correlation integral of the probability distribution is

$$\begin{aligned} C_m(r) &= \int_{-r}^{r} \int_{-r}^{r} p(x_1, \ldots, x_D | c_M)\; dx_1 \ldots dx_D \\ &= \operatorname{erf}\left(\frac{r}{\sqrt{2\sigma_{1,m}^2}}\right) \cdots \operatorname{erf}\left(\frac{r}{\sqrt{2\sigma_{D,m}^2}}\right) \quad . \end{aligned} \tag{20.54}$$

A simple calculation then shows that the cluster's correlation dimension $\nu = D_2$ is

$$\begin{aligned} \nu_m &= \frac{\partial \log C_m(r)}{\partial \log r} \\ &= \sum_{d=1}^{D} \frac{1}{\operatorname{erf}\left(r \Big/ \sqrt{2\sigma_{d,m}^2}\right)} \sqrt{\frac{2}{\pi\sigma_{d,m}^2}}\; e^{-r^2/2\sigma_{d,m}^2}\; r \quad , \end{aligned} \tag{20.55}$$

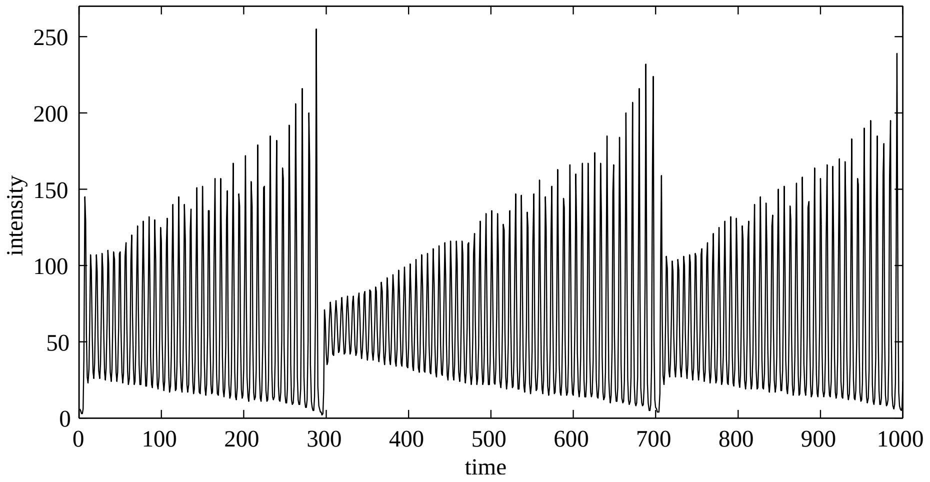


Figure 20.6. Time series of fluctuations in a laser.

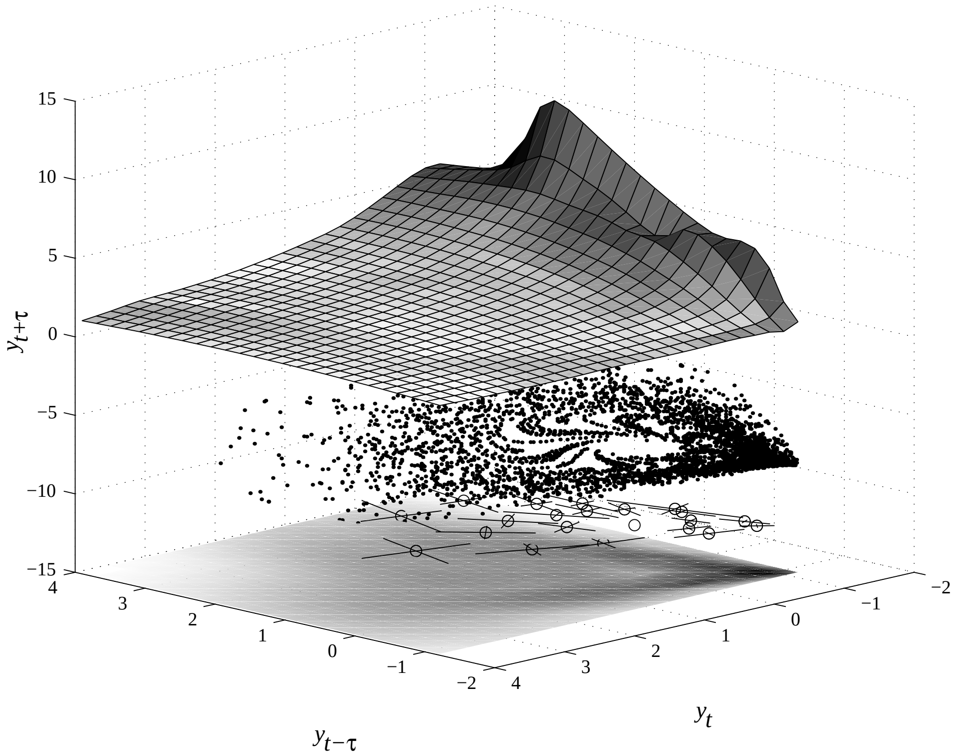


Figure 20.7. Cluster-weighted model of the time series in Figure 20.6. From bottom to top, the predicted input density estimate, the cluster means and covariances, the training data, and the conditional prediction surface with the shading showing the conditional uncertainty.

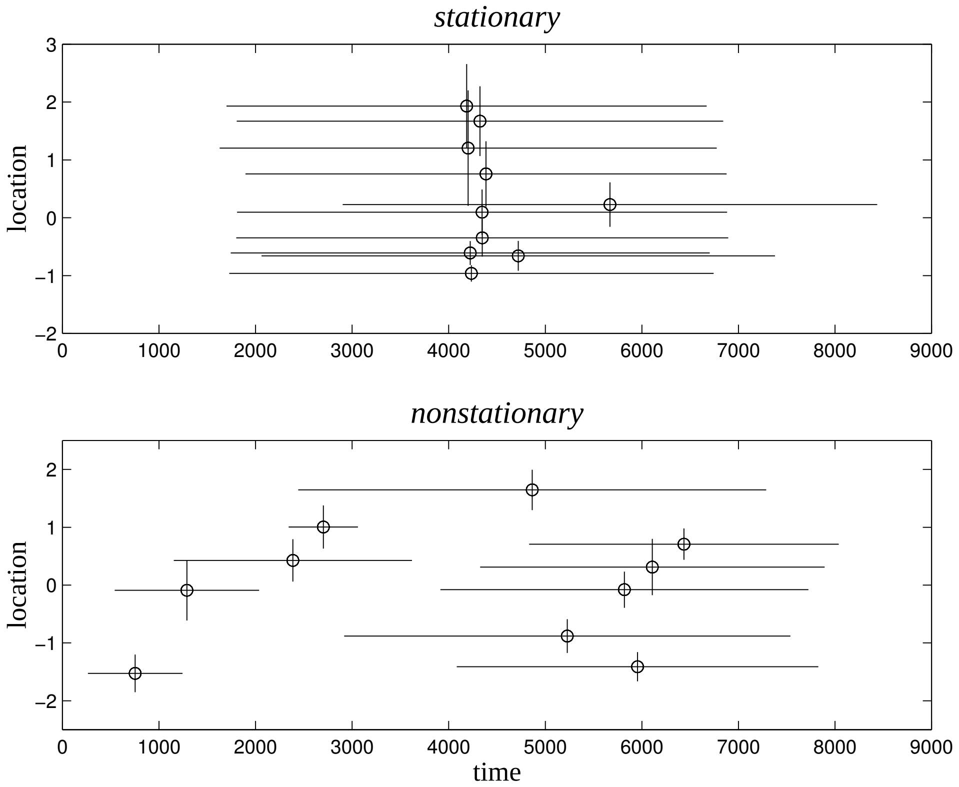


Figure 20.8. Cluster means and variances using absolute time as an input degree of freedom, shown for the time axis and first lag axis for a stationary and nonstationary time series.

and the expected dimension of the whole data set is then given by the expectation

$$\langle \nu \rangle = \sum_{m=1}^{M} \nu_m \; p(c_m) \quad . \tag{20.56}$$

This measures the average number of degrees of freedom that the system is using. Likewise, the log of the conditional output uncertainty provides an estimate of the source entropy, and hence the sum of the positive Lyapunov exponents. It's also possible to build a model recursively for on-line applications by combining new measurements with clusters that summarize old data [Gershenfeld *et al.*, 1999].

Figure 20.8 shows another example, this time with one stationary and one nonstationary series (sets A and D from [Weigend & Gershenfeld, 1993]). The absolute time has been included as one of the input degrees of freedom. For the stationary case the clusters maximize their likelihood by expanding to cover the whole data set; for the nonstationary case they shrink down to an appropriate time scale for locally building a model.

This completes the final chapter, which fittingly has drawn on lessons from many earlier parts of the book to provide useful answers to the challenging questions asked at the beginning of the chapter about coping with the breakdown of linear systems theory. We've seen techniques for allocating local models, combining them into global nonlinear functions, and describing the essential properties of a system without making restrictive assumptions about it.

20.6 SELECTED REFERENCES

[Weigend & Gershenfeld, 1993] Weigend, Andreas S., & Gershenfeld, Neil A. (eds) (1993). *Time Series Prediction: Forecasting the Future and Understanding the Past*. Santa Fe Institute Studies in the Sciences of Complexity. Reading, MA: Addison–Wesley.

The results of a comparative study in which many new and old algorithms were applied to common data sets.

[Hamilton, 1994] Hamilton, J. D. (1994). *Time Series Analysis*. Princeton, NJ: Princeton University Press.

A definitive reference for time series analysis.

20.7 PROBLEMS

(20.1) Consider the Henon map

$$\begin{aligned} x_{n+1} &= y_n + 1 - a x_n^2 \\ y_{n+1} &= b x_n \end{aligned} \tag{20.57}$$

for $a = 1.4$ and $b = 0.3$.

(a) Explore embedding by plotting

1. y_n versus x_n.
2. x_n versus n.
3. The power spectrum of x_n versus n.
4. x_{n+1} versus x_n.
5. x_{n+2} versus x_n.
6. $x_{n+1} + y_{n+1}$ versus $x_n + y_n$.

(b) Estimate as many of the nonlinear system properties as you can, such as the embedding dimension, attractor dimension, Lyapunov exponents, and source entropy.