Multiscale Analysis of Complex Time Series: Integration of Chaos and Random Fractal Theory, and Beyond
by Jianbo Gao, Yinhe Cao, Wen-wen Tung and Jing Hu

CHAPTER 13

CHAOTIC TIME SERIES ANALYSIS

In Chapter 2, we explained the essence of chaotic dynamics. In practice, one may only be able to measure a scalar time series, $x(t)$, which may be a function of the variables $V = (v_1, v_2, \cdots, v_n)$ describing the underlying dynamics (i.e., $dV/dt = f(V)$). If the dynamics are chaotic, then one can surely expect $x(t)$ to be complicated, resembling a random signal when the time scale examined is not too short. From $x(t)$, how much can we learn about the dynamics of the system? To gain insights into this fundamental problem, in Sec. 13.1 we explain how a suitable phase space can be constructed from $x(t)$ so that the dynamics of the system can be conveniently studied. This geometrical approach is among the most important methods developed in the study of chaotic dynamics. Using the problem of defending against Internet worms, we shall show that even if a dataset is not chaotic, this approach may still yield considerable insights into the complexity of the data and thus greatly simplify the problem of pattern classification. In Sec. 13.2, we discuss three most important measures for characterizing chaotic dynamics: dimension, Lyapunov exponents, and the Kolmogorov-Sinai (KS) entropy. We shall also explain how they can be computed from the $x(t)$ time series. In Sec. 13.3, we discuss a dynamical test for low-dimensional chaos. As we shall see, to determine whether a time series is chaotic or random, it is critical to incorporate the concept of scale. This point is further emphasized in Sec. 13.4, where we shall see that even

if sea clutter is not chaotic, the concept of scale can uncover nonlinear structures in the data and thus aid in detecting targets within sea clutter. Our exploration of this topic will be picked up again in Chapter 15 and greatly deepened.

13.1 PHASE SPACE RECONSTRUCTION BY TIME DELAY EMBEDDING

Given a scalar time series data $x(t)$, can we determine the dynamics of the system without reference to other variables describing it? The answer is yes; it involves constructing a suitable phase space from $x(t)$. This procedure was first introduced by Packard et al. [334], and was given a rigorous mathematical basis by Takens [421], Mane [298], and more recently by Sauer et al. [385].

13.1.1 General considerations

Suppose that a dynamical system is described by n first-order ordinary differential equations (ODEs). The dynamical system can be equivalently described by a single ODE involving terms $d^n x/dt^n$, $d^{n-1}x/dt^{n-1}$, etc. It is then clear that one way to construct a phase space is to estimate the derivatives of $x(t)$ by finite differences:

$$\frac{dx}{dt} \approx \frac{x(t+\Delta t) - x(t)}{\Delta t},$$

$$\frac{d^2x}{dt^2} \approx \frac{x(t+2\Delta t) - 2x(t+\Delta t) + x(t)}{\Delta t^2},$$

and so on, where Δt is the sampling time for $x(t)$. Unfortunately, in practice, derivatives, especially high-order ones, are quite noisy. A better approach is to develop vectors of the form

$$V_i = [x(i), x(i+L), ..., x(i+(m-1)L)], \tag{13.1}$$

where m is called the embedding dimension and L the delay time. More explicitly, we have

$$\begin{aligned}
V_1 &= [x(t_1), x(t_1+\tau), x(t_1+2\tau), ..., x(t_1+(m-1)\tau], \\
V_2 &= [x(t_2), x(t_2+\tau), x(t_2+2\tau), ..., x(t_2+(m-1)\tau], \\
&\vdots \\
V_j &= [x(t_j), x(t_j+\tau), x(t_j+2\tau), ..., x(t_j+(m-1)\tau], \\
&\vdots
\end{aligned}$$

where $t_{i+1} - t_i = \Delta t$ and $\tau = L\Delta t$. This procedure then defines a mapping (i.e., dynamics),

$$V_{n+1} = M(V_n). \tag{13.2}$$

Under the assumption that the dynamics of the system can be described by an attractor with boxing counting dimension D_F (to be defined in the next section), it can be proven that when $m > 2D_F$, the dynamics of the original system are topologically equivalent to those described by Eq. (13.2). When this is the case, the delayed reconstruction is called embedding. The basic ideas behind this fundamental theorem are that (1) given an initial condition, the solution to a set of ODEs is unique, and (2) the trajectory in the phase space does not intersect with itself. When m is not large enough, however, self-intersection may occur.

In practice, m and L have to be chosen properly. This is the issue of optimal embedding. Before taking on this issue, we consider an interesting application.

13.1.2 Defending against network intrusions and worms

Enterprise networks are facing ever-increasing security threats from distributed denial of service (DDoS) attacks, worms, viruses, intrusions, Trojans, port scans, and network misuses, and thus effective monitoring approaches to quickly detect these activities are greatly needed. Traffic-based approaches are among the most favored, because they can provide zero-hour protection against network threats if they can be executed quickly enough. To illustrate the usefulness of the geometrical approach based on the time delay embedding technique, we briefly discuss worm detection in this section.

An Internet worm is a self-propagating program that automatically replicates itself to vulnerable systems and spreads across the Internet. Most deployed worm-detection systems are signature-based. They look for specific byte sequences (called attack signatures) that are known to appear in the attack traffic. Conventionally, the signatures are manually identified by human experts through careful analysis of the byte sequence from captured attack traffic. To appreciate what attack signatures may look like, Fig. 13.1 shows two traffic traces, one normal, another for a worm. It is observed that near the left lower corner, the worm traffic is very different from the normal traffic. For this specific worm traffic, the signature sequence is indeed located there. For other worm traffic traces, the signature sequence can appear elsewhere.

In what sense may a phase space–based approach be useful? The rationale is that the signature sequence of a specific worm will occupy specific regions in the phase space. To see if this is the case, Fig. 13.2 shows the two-dimensional phase diagrams for the normal and worm traffic traces with delay time L being 1. Note that, visually, the phase diagrams remain the same when L is changed. We observe that the phase diagrams define a few threshold values. One is $TH_1 \approx 30$; another is $TH_2 \approx 120$. The exact values for TH_1 and TH_2 can be readily found by calculating the gradients of the distributions of the points in the phase plane. The phase diagrams for the normal and worm traffic differ significantly in four regions:

1. Lower left corner defined by $x(n) < TH_1,\ x(n+1) < TH_1$.

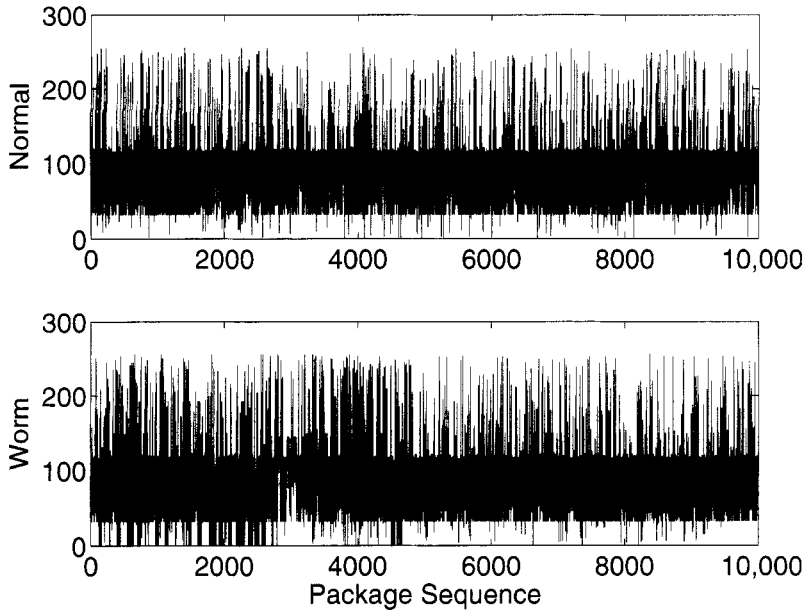

Figure 13.1. Normal and worm traffic traces.

2. Upper right corner defined by $x(n) > TH_2,\ x(n+1) > TH_2$.
3. Left vertical stripe defined by $x(n) < TH_1,\ x(n+1) > TH_2$.
4. Lower horizontal stripe defined by $x(n) > TH_2,\ x(n+1) < TH_1$.

Due to symmetry, regions 3 and 4 can be combined. Based on the percentage of the number of points in those regions, one can define three simple indices to distinguish normal and worm traffic. One result is shown in Fig. 13.3, where we observe that the histograms completely separate normal and worm data. Therefore, the accuracy is 100%. More importantly, this method is much faster than other methods, such as expectation maximization (EM) or hidden Markov model (HMM)-based approaches.

It is interesting to note that conventionally, a detection problem such as the one discussed here requires extensive training. This is no longer needed when one uses a phase space-based approach because this approach identifies a subspace that contains the signature sequence of the worm. Since the subspace depends on the rules (or dynamics) that have generated the worm signature sequence, but not on when the worm attack signatures are present, it may be appropriate to call the subspace an invariant subspace of the specific worm. While one might be concerned that byte sequences due to other network activities might occupy similar phase space regions, there is an effective solution: enlarge the embedding dimension so that the worm signature sequence can be located more precisely in the high-dimensional

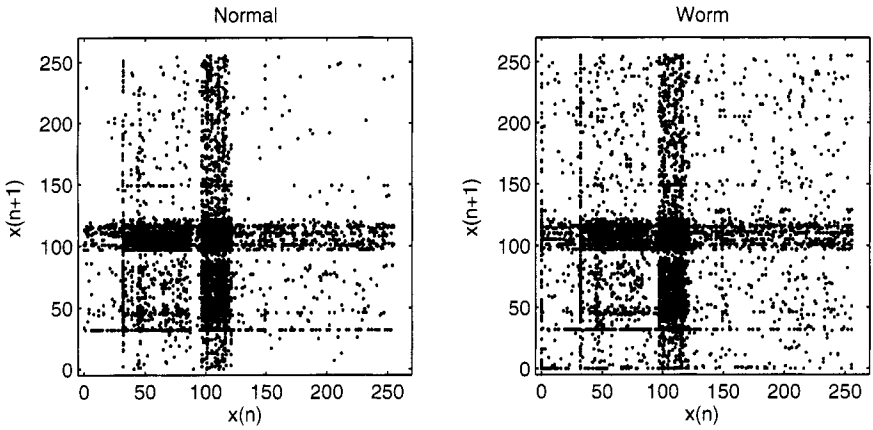

Figure 13.2. Two-dimensional phase diagrams for normal and worm traffic traces.

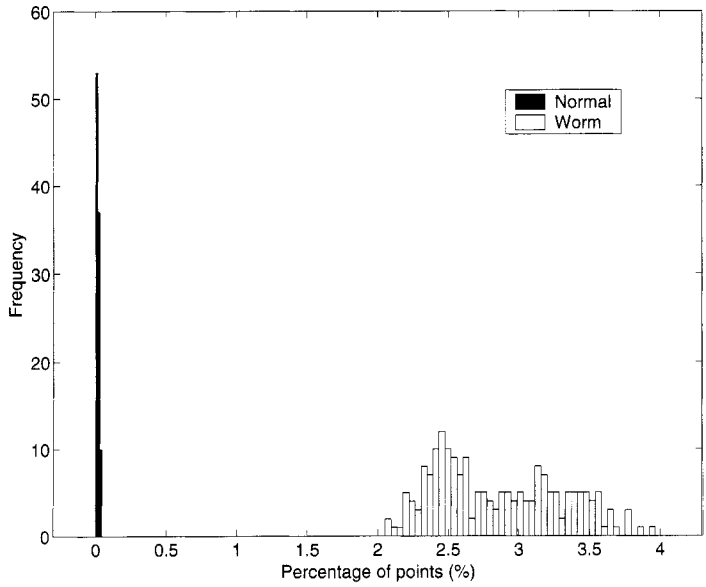

Figure 13.3. Histograms for an index based on the percentage of the number of points in region 1. 100 normal trace data and 200 worm data are used in the evaluation.

phase space. The effectiveness of this approach lies in the fact that the chance for the small phase space region occupied by the signature sequence of the worm to be shared by byte sequences of other network activities is very small.

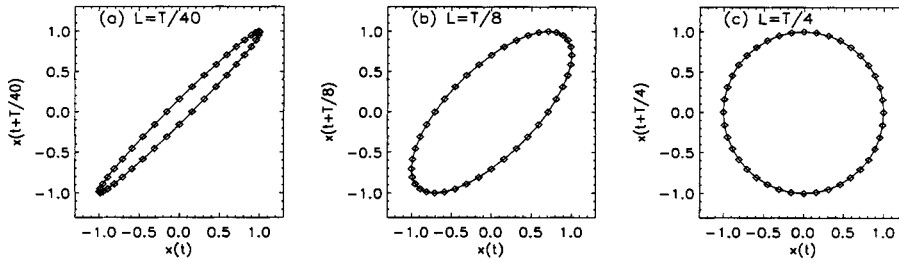

Figure 13.4. Embedding of the harmonic oscillator.

13.1.3 Optimal embedding

First, we examine a simple example, the harmonic oscillator. Without loss of generality, the dynamics of the harmonic oscillator can be described by

$$\frac{d^2x}{dt^2} = -\omega x.$$

Equivalently, they can be described by

$$\frac{dx}{dt} = y, \ \frac{dy}{dt} = -\omega x,$$

and the general solution is

$$x(t) = A\cos(\omega t + \phi_0), \ y(t) = A\sin(\omega t + \phi_0).$$

Here, the phase space is a plane spanned by x and y. Suppose only $x(t)$ is measured. If we embed $x(t)$ in a two-dimensional space, then $V(t) = [x(t), x(t+\tau)]$. Figure 13.4 shows embeddings with $\tau = T/40$, $T/8$, $T/4$, where $T = 2\pi/\omega$ is the period of the oscillation. It is clear that when $\tau = T/4$, the reconstructed phase space is identical to the original phase plane. In this simple example, the minimal embedding dimension is 2, and the optimal delay time is 1/4 of the period. This specific choice of the delay time ensures that the motion in the reconstructed phase plane is the most uniform. The magnitude of the phase velocity is everywhere the same in the reconstructed phase plane of Fig. 13.4(c) but not in those of Figs. 13.4(a,b).

We now discuss the issue of optimal embedding. First, we make a few general comments. (1) The time span represented by the embedding vector V (Eq. (13.1)) is $(m-1)\tau$. It is called the embedding window and quite often is more important than m or τ alone. However, we shall focus on the selection of m and τ, since $(m-1)\tau$ can be easily calculated once m and τ are determined. (2) As we have remarked, when m is too small, the trajectory in the reconstructed phase space may intersect with itself. This violates the uniqueness of a solution described by a set of ODEs. Therefore, m should not be too small. However, with finite data, especially when

data contain noise, m should not be too large either. (3) As illustrated in Fig. 13.4(a), when the delay time τ is small, the reconstructed phase diagram is clustered around the diagonal. This is true in an arbitrary space R^m when the embedding window is too small. Therefore, τ should not be too small. However, τ should not be too large either, since when τ is very large, successive elements of the reconstructed vector may already be almost independent, making the deterministic structure of the data difficult to assess. Below we discuss five different methods.

(1) Visual inspection: It is often instructive to check the data in a two-dimensional plane and see whether the chosen delay time makes the plot stretch in the phase plane uniformly, i.e., whether it is similar to that shown in Fig. 13.4(c). This is a qualitative method, however. Furthermore, no information on the embedding dimension m can be obtained. When one gradually increases m, it is better to decrease the delay time so that the embedding window $(m-1)\tau$ is kept approximately constant or increases more slowly than m.

(2) Determining τ based on the autocorrelation function: Empirically, it has been found that the time corresponding to the first zero of the autocorrelation function (Eq. (3.31)) of the signal is often a good estimate for τ. As with the first method, no information on m can be obtained.

(3) Determining τ based on mutual information: A refinement of the second method is based on mutual information. Let the data be partitioned into a number of bins. Denote the probability that the signal assumes a value inside the ith bin by p_i, and let $p_{ij}(\tau)$ be the probability that $x(t)$ is in bin i and $x(t+\tau)$ is in bin j. Then the mutual information for time delay τ is

$$I(\tau) = \sum_{i,j} p_{ij}(\tau) \ln p_{ij}(\tau) - 2\sum_i p_i \ln p_i. \tag{13.3}$$

In the special case $\tau = 0$, $p_{ij} = p_i \delta_{ij}$, and I yields the Shannon entropy of the data distribution. When τ is large, $x(t)$ and $x(t+\tau)$ are independent and p_{ij} factorizes to $p_i p_j$; therefore, $I \approx 0$. It has been found that a good τ corresponds to the first minimum of $I(\tau)$, when $I(\tau)$ has a minimum. In practice, when calculating $I(\tau)$, it may be advantageous to use equal-probability bins instead of equal-size bins, i.e., $p_i = 1/n = const$, when n is the number of bins used to partition the data. It should be emphasized that $p_{ij}(\tau)$ is not a constant.

(4) False nearest neighbor method: This is a geometrical method. Consider the situation in which an m_0-dimensional delay reconstruction is embedding but an (m_0-1)-dimensional reconstruction is not. Passing from $m_0 - 1$ to m_0, self-intersection in the reconstructed trajectory is eliminated. This feature can be quantified by the sharp decrease in the number of nearest neighbors when m is increased from $m_0 - 1$ by 1. Therefore, the optimal value of m is m_0. More precisely, for each reconstructed vector $V_i^{(m)} = [x(t_i), x(t_i+\tau), x(t_i+2\tau), \cdots, x(t_i+(m-1)\tau)]$, its nearest neighbor $V_j^{(m)}$ is found (for clarity, the superscript (m) is introduced

to explicitly indicate that this is an m-dimensional reconstruction). If m is not large enough, then $V_j^{(m)}$ may be a false neighbor of $V_i^{(m)}$. If embedding can be achieved by increasing m by 1, then the embedding vectors become $V_i^{(m+1)} = [x(t_i), x(t_i+\tau), x(t_i+2\tau), \cdots, x(t_i+(m-1)\tau, x(t_i+m\tau)] = [V_i^{(m)}, x(t_i+m\tau)]$ and $V_j^{(m+1)} = [V_j^{(m)}, x(t_j + m\tau)]$, and they will no longer be close neighbors. Instead, they will be far apart. The criterion for optimal embedding is then

$$R_f = \frac{|x(t_i + m\tau) - x(t_j + m\tau)|}{||V_i^{(m)} - V_j^{(m)}||} > R_T, \tag{13.4}$$

where R_T is a heuristic threshold value. Abarbanel [1] recommends $R_T = 15$.

After m is chosen, τ can be found by minimizing R_f.

While this method is intuitively appealing, we should point out that it works less effectively in the noisy case. Partly, this is because the concept of nearest neighbors is not well defined when there is noise.

(5) Time-dependent exponent curves: This is a dynamical method developed by Gao and Zheng [178,179], and provides another convenient means of quantifying the effect of self-intersection when an m-dimensional delay reconstruction is not an embedding. Let the reconstructed trajectory be denoted by $V_1^{(m)}, V_2^{(m)}, \cdots$. Assume that $V_i^{(m)}$ and $V_j^{(m)}$ are false neighbors. It is unlikely that points $V_{i+k}^{(m)}$, $V_{j+k}^{(m)}$, where k is called the evolution time, will continue to be close neighbors. That is, the separation between $V_{i+k}^{(m)}$ and $V_{j+k}^{(m)}$ will be much larger than that between $V_i^{(m)}$ and $V_j^{(m)}$ if the delay reconstruction is not an embedding. The measure proposed by Gao and Zheng is

$$\Lambda(m, L, k) = \left\langle \ln \left(\frac{||V_{i+k} - V_{j+k}||}{||V_i - V_j||} \right) \right\rangle, \tag{13.5}$$

where, for ease of later presentation, the superscript (m) in the reconstructed vectors has been dropped. The angle brackets denote the ensemble average of all possible (V_i, V_j) pairs satisfying the condition

$$\epsilon_i \leq ||V_i - V_j|| \leq \epsilon_i + \Delta\epsilon_i, \quad i = 1, 2, 3, \cdots, \tag{13.6}$$

where ϵ_i and $\Delta\epsilon_i$ are prescribed small distances. Geometrically, a pair of ϵ_i and $\Delta\epsilon_i$ define a shell, with the former being the diameter of the shell and the latter the thickness of the shell. $\Delta\epsilon_i$ is not necessarily a constant. Since the computation is carried out for a sequence of shells, the effect of noise can be largely eliminated.

Gao and Zheng suggest that for a fixed small k, an optimal m is such that $\Lambda(m, L, k)$ no longer decreases much when further increasing m. After m is chosen, L can be selected by minimizing $\Lambda(m, L, k)$. This optimization procedure ensures that the motion in the reconstructed phase space is the *most uniform*, like that shown in Fig. 13.4(c). It is found that for model chaotic systems, the embedding parameters

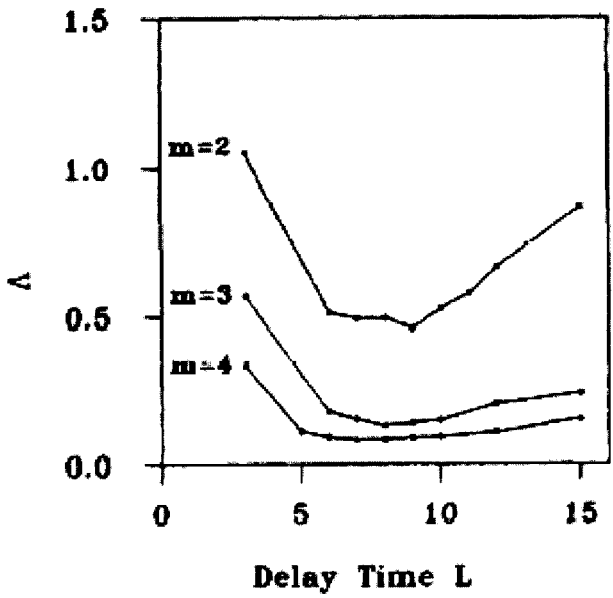

Figure 13.5. $\Lambda(L)$ vs. L for $k = 9\delta t$ and a few different m for the Rossler attractor, where $\delta t = \pi/25$ is the sampling time. 2000 points were used in the computation (adapted from Gao and Zheng [179]).

obtained by this method are the same as those obtained by the false nearest neighbor method. An example for the Rossler attractor,

$$\begin{aligned} dx/dt &= -(y+z), \\ dy/dt &= x + ay, \\ dz/dt &= b + z(x-c), \end{aligned} \tag{13.7}$$

where $a = 0.15$, $b = 0.2$, and $c = 10.0$, is shown in Fig. 13.5, where we observe that the optimal embedding parameters are $m = 3$, $L = 8$. It should be noted that the method works well on short, noisy time series because of the introduction of a series of shells instead of a small ball.

Finally, we note that $\Lambda(k)$ is related to the largest positive Lyapunov exponent. We shall have more to say on this later.

13.2 CHARACTERIZATION OF CHAOTIC ATTRACTORS

In Sec. 2.2, we briefly explained the essence of chaotic dynamics. In this section, we introduce three important measures used to characterize chaotic attractors. At this point, it would be useful to review the first few pages of Secs. 2.1 and 2.2.

13.2.1 Dimension

Dimension is a geometrical quantity characterizing the minimal number of variables needed to fully describe the dynamics of a motion. In Sec. 2.2, we explained that typically a chaotic attractor is a fractal. There are many ways to define the dimensions of a chaotic attractor. One dimension is called the capacity dimension or the box-counting dimension. It is denoted as D_F or D_0 (the meaning of the subscript 0 will be clear soon). D_0 is defined as follows: Partition the phase space containing the attractor into many cells of linear size ϵ. Denote the number of nonempty cells by $n(\epsilon)$. Then

$$n(\epsilon) \sim \epsilon^{-D_0}, \quad \epsilon \to 0.$$

Note that this equation is identical to Eq. (2.2).

The concept of the box-counting dimension can be generalized to obtain a sequence of dimensions called the generalized dimension spectrum. This is obtained by assigning a probability p_i to the ith nonempty cell. One simple way to calculate p_i is by using n_i/N, where n_i is the number of points within the ith cell and N is the total number of points on the attractor. Let the number of nonempty cells be n. Then

$$D_q = \frac{1}{q-1} \lim_{\epsilon \to 0} \left(\frac{\log \sum_{i=1}^{n} p_i^q}{\log \epsilon} \right), \tag{13.8}$$

where q is real. In general, D_q is a nonincreasing function of q. D_0 is simply the box-counting or capacity dimension, since $\sum_{i=1}^{n} p_i^q = n$. D_1 gives the information dimension D_I,

$$D_I = \lim_{\epsilon \to 0} \frac{\sum_{i=1}^{n} p_i \log p_i}{\log \epsilon}. \tag{13.9}$$

Typically, D_I is equal to the pointwise dimension α defined as

$$p(l) \sim l^{\alpha}, \quad l \to 0, \tag{13.10}$$

where $p(l)$ is the measure (i.e., probability) for a neighborhood of size l centered at a reference point. D_2 is called the correlation dimension.

The D_q spectrum is a multifractal characterization. An alternative multifractal formulation is to use singular measures. The basic idea is that a chaotic attractor is comprised of many interwoven fractals, each with a different fractal dimension. Let $f(\alpha)$ be the dimension of points with pointwise dimension α. Then we have

$$D_q = \frac{1}{q-1}[q\alpha(q) - f(\alpha(q))]. \tag{13.11}$$

Differentiating Eq. (13.11), we find

$$\alpha = \frac{d}{dq}[(q-1)D_q] \tag{13.12}$$

and then

$$f(\alpha) = (1-q)D_q + q\alpha\,. \tag{13.13}$$

It is clear that the D_q and $f(\alpha)$ spectra give the same amount of information.

In practice, when the dimension of the phase space is high and the length of the data is not very great, calculating dimension by partitioning the phase space into small boxes is not an efficient method. The practical algorithm is the Grassberger-Procaccia algorithm. It involves computing the correlation integral

$$C(\epsilon) = \lim_{N\to\infty} \frac{1}{N^2} \sum_{i,j=1}^{N} H(\epsilon - ||V_i - V_j||), \tag{13.14}$$

where V_i and V_j are points on the attractor, $H(y)$ is the Heaviside function (1 if $y \geq 0$ and 0 if $y < 0$), and N is the number of points randomly chosen from the entire dataset. The Heaviside function simply counts the number of points within the radius ϵ of the points denoted by V_i, and $C(\epsilon)$ gives the average fraction of points within a distance of ϵ. One then checks the following scaling behavior:

$$C(\epsilon) \sim \epsilon^{D_2}, \quad as \quad \epsilon \to 0. \tag{13.15}$$

Ding et al. [107] have shown that the dimensions calculated by the Grassberger-Procaccia algorithm and that defined by Eq. (13.8) (with $q = 2$) are equivalent.

When calculating the correlation integral, one may compute pairwise distances, excluding points V_i and V_j that are too close in time (i.e., i and j are too close). A rule of thumb suggested by Theiler [430] is that

$$|i - j| > w,$$

where w may be chosen as a decorrelation time. Gao and Zheng [179] pointed out that when V_i and V_j are close in time, they may be on the same orbit. The dimension corresponding to such tangential motion is 1, while the Lyapunov exponent is 0. This point will be made clearer later.

As an example, we compute the dimension for the Lorenz attractor:

$$\begin{aligned} dx/dt &= -16(x-y) + D\eta_1(t), \\ dy/dt &= -xz + 45.92x - y + D\eta_2(t), \\ dz/dt &= xy - 4z + D\eta_3(t). \end{aligned} \tag{13.16}$$

For later convenience, we have added independent Gaussian noise forcing terms $\eta_i(t)$, $i = 1, 2, 3$ with mean 0 and variance 1. D characterizes the strength of noise. When $D = 0$, the system is clean. The top plot of Fig. 13.6 shows the correlation integral for the $x(t)$ time series of the Lorenz attractor, where the curves from top to bottom correspond to $m = 4, 6, \cdots, 12$. The bottom plot of Fig. 13.6 is the so-called plateau plot of the correlation integrals, where we observe that a plateau common to different m occurs at 2.05. This is the correlation dimension for the Lorenz attractor.

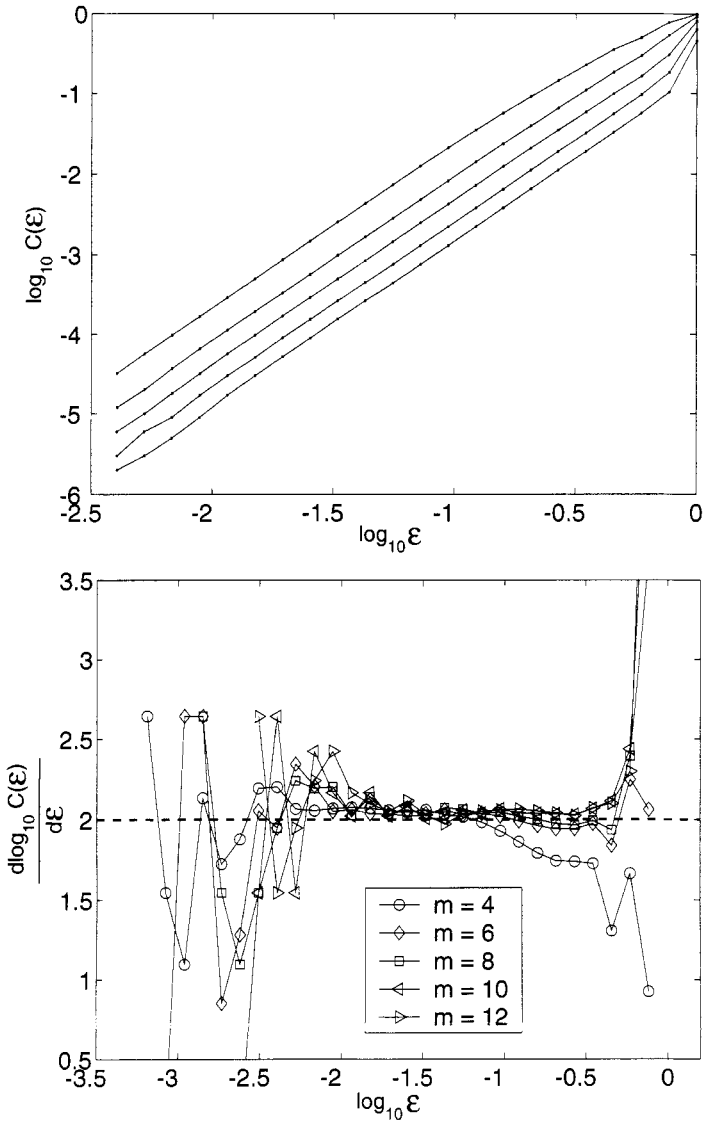

Figure 13.6. The correlation integral (top) and the correlation dimension (bottom) for the Lorenz attractor. The sampling time is 0.06. 10000 points were used in the computation.

13.2.2 Lyapunov exponents

General considerations: Lyapunov exponents are dynamical quantities. In Sec. 2.2, we introduced the concept of Lyapunov exponents by considering the evolution of an infinitesimal line segment. We now discuss the entire spectrum of the Lyapunov exponents by considering the evolution of an infinitesimal ball of radius dr. Fig-

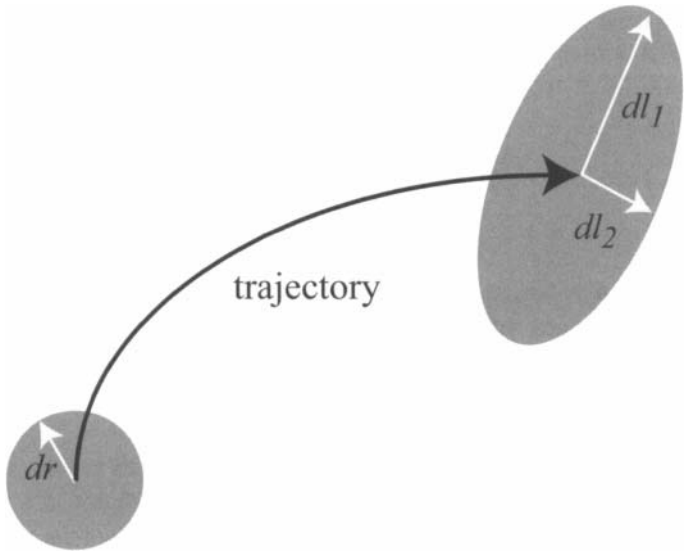

Figure 13.7. Calculation of the Lyapunov exponents in the two-dimensional case: evolution of a small circle to an ellipsoid.

ure 13.7 schematically shows the evolution of an infinitesimal ball to an ellipsoid. Let $l_i(t)$ be the ith principal axis of the ellipsoid at time t. Generalizing Eq. (2.4), we can write

$$l_i(t) \sim dr \cdot e^{\lambda_i t}.$$

More formally, we have

$$\lambda_i = \lim_{dr \to 0,\, t \to \infty} \frac{1}{t} \ln \frac{l_i(t)}{dr}. \tag{13.17}$$

The Lyapunov exponents are conventionally listed in descending order: $\lambda_1 \geq \lambda_2 \geq \lambda_3 \cdots$. When λ_1 is positive, the dynamics of the system are said to be chaotic. A dynamical system may have multiple positive Lyapunov exponents.

We now examine the general properties of the Lyapunov exponents for an n-dimensional system described by

$$dV/dt = f(V),$$

where $V = (v_1, v_2, \cdots, v_n)$. Let the volume of a small ball at time $t = 0$ be denoted by $B(0)$. At time t, the volume becomes

$$B(t) = B(0) e^{(\sum_{i=1}^{n} \lambda_i)t}. \tag{13.18}$$

This is the solution to

$$dB(t)/dt = \Big(\sum_{i=1}^{n} \lambda_i\Big) B(t). \tag{13.19}$$

On the other hand, using the divergence theorem, the rate of change of the phase volume can be written as

$$dB(t)/dt = \bigtriangledown \cdot fB(t) \,. \tag{13.20}$$

Therefore,

$$\sum_{i=1}^{n} \lambda_i = \bigtriangledown \cdot f \,. \tag{13.21}$$

A dynamical system is said to be conservative if $\bigtriangledown \cdot f = 0$ and dissipative if $\bigtriangledown \cdot f < 0$. Furthermore, a continuous chaotic dynamical system must have a zero Lyapunov exponent, corresponding to the situation in which a principal axis $l_i(t)$ is parallel to the trajectory. In the chaotic Lorenz system of Eq. (13.16), for example, the three Lyapunov exponents are $(1.50, 0, -22.46)$.

We now discuss the relation between the Lyapunov exponents and the dimension. Assume there are j positive Lyapunov exponents. At time 0, let us consider a $j+1$-dimensional volume of size r, where $r \to 0$. At time t, the edges of the infinitesimal volume become $re^{\lambda_i t}$, $i = 1, 2, \cdots, j+1$, where λ_{j+1} is the negative Lyapunov exponent with the smallest magnitude. Now let us cover this volume by boxes of size $\epsilon = re^{\lambda_{j+1} t}$. It is easy to see that we need

$$N(\epsilon) = \Pi_{i=1}^{j}(e^{\lambda_i t}/e^{\lambda_{j+1} t}) = e^{\sum_{i=1}^{j}(\lambda_i - \lambda_{j+1})t} \sim \epsilon^{-D_L}$$

boxes to cover it, where the dimension is now indexed by L and is called the Lyapunov dimension or Kaplan-Yorke dimension. Therefore,

$$D_L = j - \sum_{i=1}^{j} \lambda_i/\lambda_{j+1} = j + \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_j}{|\lambda_{j+1}|} \,. \tag{13.22}$$

Note that $e^{\lambda_i t}$, $i > j+1$ converge to 0 more rapidly than $e^{\lambda_{j+1} t}$, since λ_i, $i > j+1$ are more negative than λ_{j+1}. Hence, those Lyapunov exponents do not have any effect on the dimension.

Numerical computations: There are a few straightforward ways to calculate the largest positive Lyapunov exponent from a time series. In order of increasing sophistication, we describe three of them here.

(1) Wolf et al.'s algorithm [478]: The basic idea is to select a reference trajectory and follow the divergence of a neighboring trajectory from it. Let the spacing between the two trajectories at time t_i and t_{i+1} be d_i' and d_{i+1}, respectively. The rate of divergence of the trajectory over a time interval of $t_{i+1} - t_i$ is then

$$\frac{\ln(d_{i+1}/d_i')}{t_{i+1} - t_i} \,.$$

To ensure that the separation between the two trajectories is always small, when d_{i+1} exceeds certain threshold value, it has to be renormalized: a new point in the

direction of the vector of d_{i+1} is picked up so that d'_{i+1} is very small compared to the size of the attractor. After n repetitions of stretching and renormalizing the spacing, one obtains the following formula:

$$\lambda_1 = \sum_{i=1}^{n-1} \left[\frac{t_{i+1} - t_i}{\sum_{i=1}^{n-1}(t_{i+1} - t_i)} \right] \left[\frac{\ln(d_{i+1}/d'_i)}{t_{i+1} - t_i} \right] = \frac{\sum_{i=1}^{n-1} \ln(d_{i+1}/d'_i)}{t_n - t_1} . \quad (13.23)$$

Note that this algorithm assumes but does not verify exponential divergence. In fact, the algorithm can yield a positive value of λ_1 for any type of noisy process so long as all the distances involved are small. As pointed out by Gao and Zheng, the reason for this is that when d'_i is small, evolution would move d'_i to the most probable spacing. Then, d_{i+1}, being in the middle step of this evolution, will be larger than d'_i; therefore, a quantity calculated based on Eq. (13.23) will be positive. This argument makes it clear that the algorithm cannot distinguish chaos from noise. We will come back to this issue later.

(2) Rosenstein et al.'s [372] and Kantz's [248] algorithm: In this method, one first chooses a reference point and finds its ϵ-neighbors V_j. One then follows the evolution of all these points and computes an average distance after a certain time. Finally, one chooses many reference points and takes another average. Following the notation of Eq. (13.5), these steps can be described by

$$\Lambda(k) = \left\langle \ln \langle \|V_{i+k} - V_{j+k}\| \rangle_{average\ over\ j} \right\rangle_{average\ over\ i} , \quad (13.24)$$

where V_i is a reference point and V_j are neighbors to V_i, satisfying the condition $\|V_i - V_j\| < \epsilon$. If $\Lambda(k) \sim k$ for a certain intermediate range of k, then the slope is the largest Lyapunov exponent (The reason for the intermediate range of k will be explained when we discuss the third method.)

While in principle this method can distinguish chaos from noise, with finite noisy data it may not function as desired. One of the major reasons is that in order for the *average over* j to be well defined, ϵ has to be small. In fact, sometimes the ϵ-neighborhood of V_i is replaced by the nearest neighbor of V_i. For this reason, the method cannot handle short, noisy time series well.

(3) Gao and Zheng's method [178–180]: This method contains three basic ingredients: Eq. (13.5), Eq. (13.6), and the condition

$$|i - j| > w. \quad (13.25)$$

When the Lyapunov exponent is calculated, it is usually assumed that the embedding parameters have been properly chosen. In the rest of the book, we shall simply write $\Lambda(k)$ instead of $\Lambda(m, L, k)$. Inequality (13.25) ensures that tangential motions, corresponding to the condition that V_i and V_j follow each other along the orbit, are removed. Tangential motions contribute a Lyapunov exponent of zero and, hence, severely underestimate the positive Lyapunov exponent. An example is shown

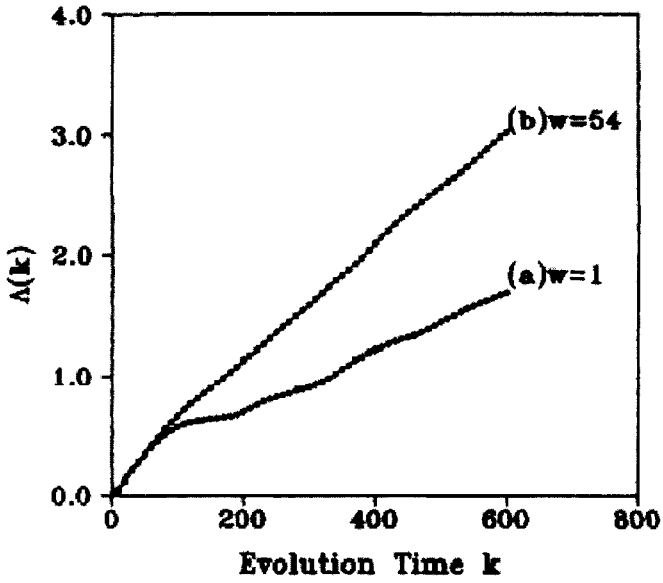

Figure 13.8. $\Lambda(k)$ vs. k curves for the Lorenz system. When $w = 1$, the slope of the curve severely underestimates the largest Lyapunov exponent. When w is increased to 54, the slope correctly estimates the largest Lyapunov exponent.

in Fig. 13.8, where we observe that $w = 1$ severely underestimates the largest Lyapunov exponent, while $w = 54$ solves the problem. In practice, w can be chosen to be larger than one orbital time, such as in the case of the Lorenz attractor and the Rossler attractor. If an orbital time cannot be defined, it can be more or less arbitrarily set to be a large integer if the dataset is not too small. Note that this parameter is closely related to the Theiler decorrelation time used to improve the calculation of correlation dimension, as explained earlier. However, relating w to autocorrelation time is a bit misleading, since the origin of this time scale is due to tangential motions along the orbit.

The condition described by Inequality (13.25) constitutes one of the major differences between this method and the method of Rosenstein et al. There are two more major differences between this method and the method of Rosenstein et al: (1) Gao and Zheng's method calculates $\Lambda(k)$ for a sequence of shells, not just the ϵ-neighbors (which is a ball). As will be explained in Sec. 13.3, this step is critical for the method to be a dynamical test for deterministic chaos. (2) Gao and Zheng's method takes the logarithm of $\|V_{i+k} - V_{j+k}\|$ before taking the average. Rosenstein et al.'s method, by contrast, takes the average over j first, then takes the logarithm, and then takes another average over i.

We will return to this method in Sec. 13.3 when we discuss tests for low-dimensional chaos.

13.2.3 Entropy

General considerations: Entropy characterizes the rate of creation of information in a system. Let us consider an imaginary dynamical system. To calculate the entropy of the motion, we can partition the phase space into small boxes of size ϵ, compute the probability p_i that box i is visited by the trajectory, and finally use the formula $I = -\sum p_i \ln p_i$. For many systems, information increases linearly with time:

$$I(\epsilon, t) = I_0 + Kt, \tag{13.26}$$

where I_0 is the initial entropy and K is the KS entropy (its more precise definition will follow shortly). Now suppose that the system is initially in a particular region of the phase space, and all initial probabilities are zero except the corresponding probability for that region, which is 1. Therefore, $I_0 = 0$.

We now consider three cases of the system: (1) deterministic, nonchaotic, (2) deterministic, chaotic, and (3) random. For case (1), during the time evolution of the system, phase trajectories remain close together. After a time T, nearby phase points are still close to each other and can be grouped in some other small region of the phase space. Therefore, there is no change in information. For case (2), due to exponential divergence, the number of phase space regions available to the system after a time T is $N \propto e^{(\sum \lambda^+)T}$, where λ^+ are positive Lyapunov exponents. Assuming that all of these regions are equally likely, the information function then becomes

$$I(T) = -\sum_{i=1}^{N} p_i(T) \ln p_i(T) = (\sum \lambda^+)T\,, \tag{13.27}$$

where it is assumed that $p_i(T) \sim 1/N$. Therefore, $K = \sum \lambda^+$. More generally, if these phase space regions are not visited with equal probability, then

$$K \leq \sum \lambda^+\,. \tag{13.28}$$

Grassberger and Procaccia, however, suggest that equality usually holds. Finally, for case (3), we can easily imagine that after a short time, the entire phase space may be visited. Therefore, $I \sim \ln N$. When $N \to \infty$, we have $K = \infty$.

In summary, K is zero for regular motions, positive and finite for chaotic motions, and infinite for random motions.

Formal definitions: First, let us precisely define the KS entropy. Consider a dynamical system with F degrees of freedom. Suppose that the F-dimensional phase space is partitioned into boxes of size ϵ^F. Suppose that there is an attractor in phase space and consider a transient-free trajectory $\vec{x}(t)$. The state of the system is now measured at intervals of time τ. Let $p(i_1, i_2, \cdots, i_d)$ be the joint probability that $\vec{x}(t = \tau)$ is in box i_1, $\vec{x}(t = 2\tau)$ is in box i_2, $\cdots$, and $\vec{x}(t = d\tau)$ is in box i_d.

The KS entropy is then

$$K = -\lim_{\tau\to 0}\lim_{\epsilon\to 0}\lim_{d\to\infty}\frac{1}{d\tau}\sum_{i_1,\cdots,i_d} p(i_1,\cdots,i_d)\ln p(i_1,\cdots,i_d)\,. \tag{13.29}$$

Alternatively, we may first introduce the block entropy:

$$H_d(\epsilon,\tau) = -\sum_{i_1,\cdots,i_d} p(i_1,\cdots,i_d)\ln p(i_1,\cdots,i_d). \tag{13.30}$$

It is on the order of $d\tau K$; then we take difference between $H_{d+1}(\epsilon,\tau)$ and $H_d(\epsilon,\tau)$ and normalize by τ:

$$h_d(\epsilon,\tau) = \frac{1}{\tau}[H_{d+1}(\epsilon,\tau) - H_d(\epsilon,\tau)]. \tag{13.31}$$

Let

$$h(\epsilon,\tau) = \lim_{d\to\infty} h_d(\epsilon,\tau). \tag{13.32}$$

It is clear that the KS entropy can also be obtained by taking proper limits in Eq. (13.32):

$$K = \lim_{\tau\to 0}\lim_{\epsilon\to 0} h(\epsilon,\tau) = \lim_{\tau\to 0}\lim_{\epsilon\to 0}\lim_{d\to\infty}\frac{1}{\tau}[H_{d+1}(\epsilon,\tau) - H_d(\epsilon,\tau)]. \tag{13.33}$$

The KS entropy can be easily extended to order-q Renyi entropies:

$$K_q = -\lim_{\tau\to 0}\lim_{\epsilon\to 0}\lim_{d\to\infty}\frac{1}{d\tau}\frac{1}{q-1}\ln\sum_{i_1,\cdots,i_d} p^q(i_1,\cdots,i_d). \tag{13.34}$$

When $q\to 1$, $K_q\to K$. As we have seen in Sec. 11.1, in the case of unequal probabilities, the Renyi entropy of order-q is a nonincreasing function of q.

Numerical calculations: There are three simple ways to calculate the KS entropy from a time series. One is to first estimate all the positive Lyapunov exponents, and then use the summation of these exponents to estimate the KS entropy. Alternatively, one can estimate K by approximating the probabilities $p(i_1,\cdots,i_m)$, as proposed by Cohen and Procaccia [80] and Eckmann and Ruelle [116]. Let the length of a time series be N, and let the m-dimensional embedding vectors be explicitly denoted by $V_i^{(m)}$, as we did in Sec. 13.1.3. For ease of interpretation, assume the delay time to be 1. Let $n_i^{(m)}(\epsilon)$ be the number of vectors $V_j^{(m)}$ satisfying $||V_j^{(m)} - V_i^{(m)}|| \le \epsilon$. Cohen and Procaccia noted that

$$C_i^{(m)}(\epsilon = n_i^{(m)}/(N-m+1)$$

approximates the probability $p(i_1,\cdots,i_m)$ for boxes of size 2ϵ. They then proposed to estimate $H_m(\epsilon)$ by

$$H_m(\epsilon) = -\frac{1}{N-m+1}\sum_i \ln C_i^{(m)}\,. \tag{13.35}$$

Then

$$K = \lim_{\tau \to 0} \lim_{\epsilon \to 0} \lim_{m \to \infty} \frac{1}{\delta t} \Big[H_{m+1}(\epsilon) - H_m(\epsilon) \Big], \tag{13.36}$$

where δt is the sampling time. Note that with the delay time $L = 1$, when the maximum norm,

$$||V_j^{(m)} - V_i^{(m)}|| = \max_{0 \le k \le m-1} |x(i+k) - x(j+k)|,$$

is used, $H_{m+1}(\epsilon) - H_m(\epsilon)$ is the logarithm of the conditional probability

$$|x(i+m) - x(j+m)| \le \epsilon,$$

given that

$$|x(i+k) - x(j+k)| \le \epsilon \text{ for } k = 0, 1, \cdots, m-1$$

averaged over i.

The third method is to approximate K by estimating K_2 through the correlation integral (defined in Eq. (13.14)):

$$C_m(\epsilon) \sim \epsilon^{D_2} e^{-m\tau K_2}, \tag{13.37}$$

where $\tau = L\delta t$ is the actual delay time. Note that $C_m(\epsilon)$ is the average of $C_i^{(m)}(\epsilon)$; therefore, it amounts to $\sum_{i_1, \cdots, i_m} p^2(i_1, \cdots, i_m)$. Equation (13.37) can also be expressed as

$$K_2 = \lim_{\tau \to 0} \lim_{\epsilon \to 0} \lim_{m \to \infty} \frac{1}{\tau} [\ln C_m(\epsilon) - \ln C_{m+1}(\epsilon)]. \tag{13.38}$$

In actual computations, due to the finite size of the data, one cannot take the limits. Instead, one focuses on scaling behavior. For example, when one works with the correlation integral, for true low-dimensional chaotic dynamics, in a plot of $\ln C_m(\epsilon)$ vs. $\ln \epsilon$ with m as a parameter, one observes a series of parallel straight lines, with the slope being the correlation dimension D_2, and the spacing between the lines estimating K_2 (where lines for larger m lie below those for smaller m). See the upper plot of Fig. 13.6. Note that τ has to be small. Otherwise, K_2 may be underestimated.

As can be seen, noisy experimental data may not possess fractal scaling behavior (e.g., defined by Eq. (13.37)) on finite scales. And one cannot take the limit of $\lim_{\epsilon \to 0}$! Then what should one do? To answer this question, a number of derivatives of the KS entropy have been proposed. They include the approximate entropy, the (ϵ, τ)-entropy, and the sample entropy. The approximate entropy amounts to Eq. (13.36) without taking the limits of $\lim_{\epsilon \to 0}$ and $\lim_{m \to \infty}$. It is usually computed using the maximum norm. The (ϵ, τ)-entropy is simply defined by Eq. (13.32). The sample entropy amounts to Eq. (13.38) without taking the limits of $\lim_{\epsilon \to 0}$ and $\lim_{m \to \infty}$. It is clear that these three entropy measures are closely related. Gaspard and Wang [183] have carefully considered the dependence of the (ϵ, τ)-entropy on

the scale ϵ. Unfortunately, it is hard to estimate these entropy measures accurately from finite data. This has motivated development of the concept of a finite-size Lyapunov exponent (FSLE). However, the standard method of calculating the FSLE still has severe limitations. To overcome these limitations, and to make the concept applicable to the study of continuous but nondifferential stochastic processes, in Chapter 15 we develop a new concept, the scale-dependent Lyapunov exponent (SDLE).

Finally, we note that recently, Costa et al. [83, 84] have developed a method, multiscale entropy analysis, by calculating the sample entropy on the original data as well as on the smoothed data defined by Eq. (8.3) and examining the dependence of the sample entropy on the block size used for smoothing.

13.3 TEST FOR LOW-DIMENSIONAL CHAOS

To determine whether a time series measured from a complex system is regular, deterministically chaotic, or random is a fundamental issue in science and engineering. Often it is assumed that a numerically estimated positive value for the Lyapunov exponent and a noninteger value for the fractal dimension are sufficient indicators of chaoticness in a time series. This assumption led to a great deal of research in many areas of the natural and social sciences as well as engineering, resulting in the conclusion that *chaos is everywhere*! Osborne and Provenzale [328, 357] observed that $1/f^{\beta}$ noise generates time series with a finite correlation dimension and converging K_2 entropy estimates. Since then, it has been realized that the many so-claimed chaotic motions are not chaotic but random. The difficulty of determining whether a complex time series is truly low-dimensional chaos or not has motivated some researchers to tackle related but less difficult problems, such as whether embedding of a time series defines directional vectors or whether the trajectory in the reconstructed phase space is continuous and smooth. Here we show that this issue can in fact be neatly handled using Gao and Zheng's method. In this section, we discuss the generic behaviors of the $\Lambda(k)$ curves for low-dimensional chaos and then consider two types of noise, iid random variables and surrogate data of low-dimensional chaotic signals. The latter are obtained by taking the Fourier transform of a chaotic signal, randomizing the phases of the Fourier transform, and then taking the inverse Fourier transform. Using the Wiener-Khintchine theorem, we see that surrogate data have the same autocorrelation function as the chaotic signal. $\Lambda(k)$ curves also have distinctive behaviors for random fractals, including $1/f^{\beta}$ processes. However, we shall postpone discussion of random fractals in Chapters 14 and 15.

To illustrate the generic behavior of the $\Lambda(k)$ curves for chaotic systems, we study the Lorenz system described by Eq. (13.16). Fig. 13.9(a) shows the $\Lambda(k)$ curves calculated from the x-component of the clean Lorenz system (i.e., $D = 0$). We observe that the curves have three characteristics:

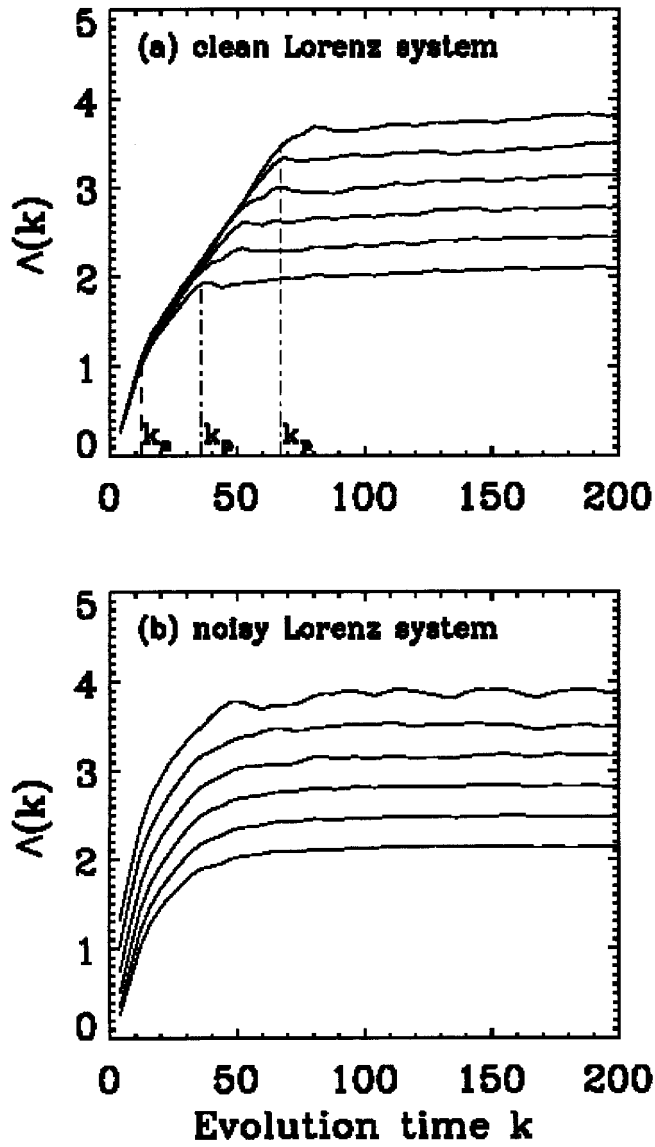

Figure 13.9. Time-dependent exponent $\Lambda(k)$ vs. evolution time k curves for (a) clean and (b) noisy Lorenz systems. Six curves, from bottom to top, correspond to shells $(2^{-(i+1)/2}, 2^{-i/2})$ with $i = 8, 9, 10, 11, 12$, and 13. The sampling time for the system is 0.03 s, and embedding parameters are $m = 4$, $L = 3$. 5000 points were used in the computation.

1. They are linearly increasing for $0 \leq k \leq k_a$.
2. They are still linearly increasing for $k_a \leq k \leq k_p$, but with a slightly different slope.
3. They are flat for $k \geq k_p$.

Feature 3 indicates that on average, the distance between embedding vectors V_{i+k} and V_{j+k} has reached the most probable separation on the attractor (e.g., the diameter of the attractor). Note that the slope of the second linearly increasing part provides an accurate estimate of the largest positive Lyapunov exponent. k_a is related to the time scale for a pair of nearby points (X_i, X_j) to evolve to the unstable manifold of V_i or V_j. It is on the order of the embedding window length, $(m-1)L$. k_p is the prediction time scale. It is longer for the $\Lambda(k)$ curves that correspond to smaller shells. The difference between the slopes of the first and second linearly increasing parts is caused by the discrepancy between the direction defined by the pair of points (V_i, V_j) and the unstable manifold of V_i or V_j. This feature was first observed by Sano et al. [384] and was used by them to improve the estimation of the Lyapunov exponent. The larger slope in the first linearly increasing part is due to superposition of multiple exponential growths, as recently pointed out by Smith et al. [405]. This region may be made smaller or can even be eliminated by adjusting the embedding parameters, such as by using a larger value for m. Note that the second linearly increasing parts of the $\Lambda(k)$ curves collapse together to form a linear envelope. This feature makes the method a direct dynamical test for deterministic chaos, since such a feature cannot be found in any nonchaotic data. As examples, we have shown the $\Lambda(k)$ for independent, uniformly distributed random variables in Fig. 13.10 and the $\Lambda(k)$ for the surrogate data of the chaotic Lorenz time series in Fig. 13.11. In those figures, we observe that the $\Lambda(k)$ curves are composed of only two parts, an increasing (and sometimes fairly linear) part for $k \leq (m-1)L$ and a flat part. The flat part again indicates that on average, the distance between embedding vectors V_{i+k} and V_{j+k} has reached the most probable separation defined by the data (now there is no longer an attractor). The first part is simply due to the fact that the initial separation between V_i and V_j is usually smaller than the most probable separation. More importantly, for noisy data, the $\Lambda(k)$ curves corresponding to different shells separate from each other, and a common envelope cannot be defined. Therefore, if one estimates the slopes as the largest positive Lyapunov exponent λ_1, then the value of λ_1 depends on which shell is used. This is a very clear indication that the data are random.

Next, we discuss noisy chaos. As expected, the behavior of the $\Lambda(k)$ curves for a noisy, chaotic system lies between that of the $\Lambda(k)$ curves for a clean, chaotic system and that of the $\Lambda(k)$ curves for white noise or for the surrogate data of a chaotic signal. This is indeed so, as can be easily observed from Fig. 13.9(b). Note that the separation is larger between the $\Lambda(k)$ curves corresponding to smaller shells. This indicates that the effect of noise on the small-scale dynamics is stronger than that on the large-scale dynamics. Also note that $k_a + k_p$ is now on the order of

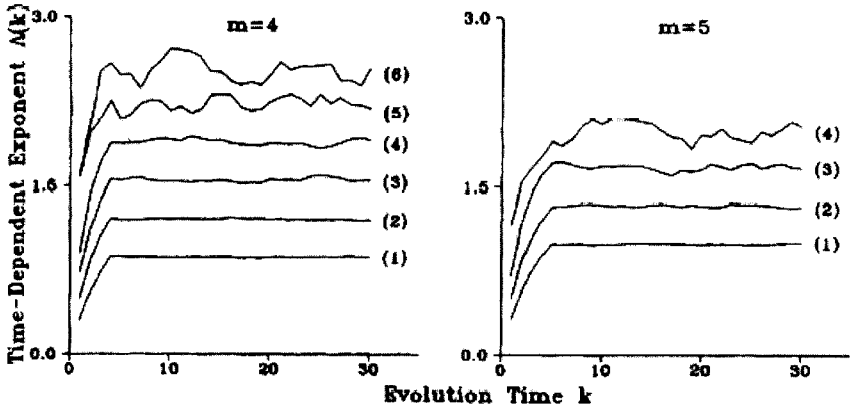

Figure 13.10. $\Lambda(k)$ curves for 6000 uniformly distributed random variables. Curves (1) to (6) correspond to shells $(2^{-(i+1)/2}, 2^{-i/2})$ with $i = 4, 5, \cdots, 9$ (when $m = 5$, curves (5) and (6) are not resolved, since there are no points in the corresponding shells). Adapted from Gao and Zheng [179].

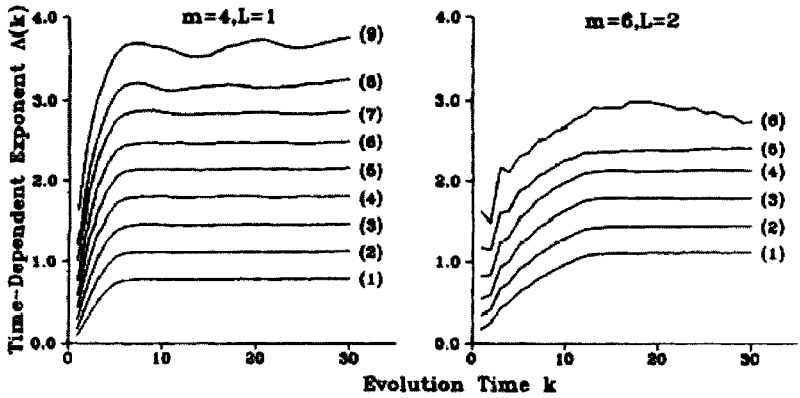

Figure 13.11. $\Lambda(k)$ curves for 6000 points of the surrogate data of the chaotic Lorenz signal. Curves (1)-(9) correspond to shells $(2^{-(i+1)/2}, 2^{-i/2})$ with $i = 4, 5, \cdots, 12$. For the case of $m = 6, L = 2$, only curves (1) to (6) are resolved. Adapted from Gao and Zheng [179].

the embedding window length and is almost the same for all the $\Lambda(k)$ curves. With stronger noise ($D > 4$), the $\Lambda(k)$ curves will be more like those for white noise.

At this point, it is pertinent to comment on why methods that do not employ shells cannot be used to reliably distinguish low-dimensional chaos from noise. The reason is as follows. Those methods amount to estimating the Lyapunov exponent by $\Lambda(k)/k$, where $\Lambda(k)$ is computed based on a single ball. The latter either

corresponds to the smallest shell or can be obtained by grouping together the series of shells in the Gao and Zheng's method. As already pointed out, for noisy data, the value of the estimated $\Lambda(k)$ will be different if the size of the ball varies. However, so long as the ball is small, $\Lambda(k)$ is typically positive. Therefore, there is always a risk of interpreting noisy data as chaotic data.

13.4 THE IMPORTANCE OF THE CONCEPT OF SCALE

In the previous section, we emphasized the importance of the concept of scale as introduced by the series of shells in Gao and Zheng's method. In this section, we further illustrate this by detecting targets within sea clutter using the largest Lyapunov exponent.

Like the study of other complex time series, determining whether sea clutter is chaotic or random has been a hot topic. Recent works demonstrating that sea clutter may not be truly chaotic are mostly based on the observation that the Lyapunov exponent, λ_c, obtained using canonical methods is similar for sea clutter and some stochastic processes that share certain features with sea clutter. Extending this idea to target detection, one can readily find that λ_c is not effective in distinguishing data with and without a target. This is evident from Figs. 13.12(a,b): λ_c for the primary target bins of most measurements are not very different from data without targets.

However, the situation changes drastically if one calculates the $\Lambda(k)$ curve for a shell of fairly large size and then estimates the slope. Denote the resulting Lyapunov exponent by λ_ϵ. The results are shown in Figs. 13.12(c,d). Now we observe that λ_ϵ is larger for the primary target bin than for data without targets. This example clearly shows the importance of incorporating the concept of scale in a measure. In fact, the concept of scale is incorporated in the $\Lambda(k)$ curves of Gao and Zheng only in a static manner. In Chapter 15, we shall see that when a measure dynamically incorporates the concept of scale, it becomes much more powerful.

13.5 BIBLIOGRAPHIC NOTES

Much of the material covered in this chapter can also be found in [1, 25, 249]. For time delay embedding, we refer to [298, 334, 385, 421]. For optimal embedding, we refer to [8, 65, 140, 255, 285]. For characterization of chaotic time series, we refer to [80, 107, 108, 116, 178–180, 203, 204, 248, 344, 372, 373, 378, 384, 478]. For approximate entropy, sample entropy, (ϵ, τ)-entropy, and multiscale entropy analysis, we refer to [83, 84, 183, 345, 367]. For distinguishing chaos from noise, we refer to [28, 104, 105, 179, 180, 182, 197, 229, 251, 256, 339, 352, 358, 380, 416, 451, 469]. For chaos in fluid flows, we refer to [417, 418]. For surrogate data, we refer to [432]. Finally, for worm detection, we refer to [423] and references therein.

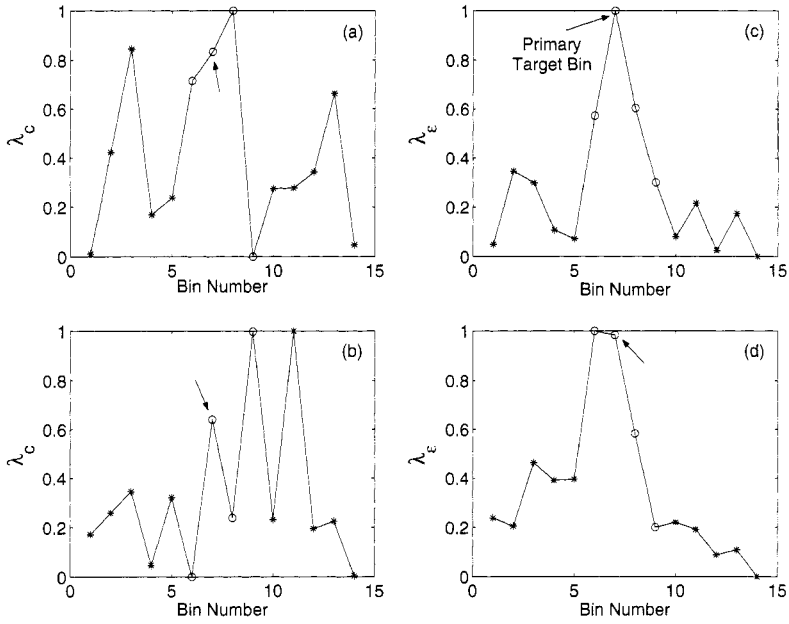

Figure 13.12. (a,b) The variations of the largest Lyapunov exponent λ_c (estimated by a canonical method) for two measurements, each with 14 range bins. Open circles denote the range bins with a target, while asterisks denote the bins without a target. The primary target bin is indicated by an arrow. (c,d) The variations of the Lyapunov exponent, denoted as λ_ϵ, corresponding to large shells for the same two measurements.

13.6 EXERCISES

1. Generate a time series from $x(t) = \sin\omega_1 t + a\sin\omega_2 t$ and ω_1/ω_2 is an irrational number by choosing appropriate parameters where a is a constant. Embed the time series to a two-dimensional phase space. Do your plots look like a torus?

2. Write a program to compute the correlation integral from the chaotic Lorenz attractor data (see Sec. A.4 of Appendix A), then compute the correlation dimension and the correlation entropy. (Hint: If you have difficulty writing the program, you may compare your code with the one listed in the book's website explained in Sec. A.4 of Appendix A.)

3. Write a program to compute the $\Lambda(k)$ curves from various data downloadable at the book's website (see Sec. A.4 of Appendix A).