Chapter 9

Information Theory in Dynamical Systems

In this chapter, we outline the strong connection between dynamical systems and a symbolic representation through symbolic dynamics. The connection between dynamical systems and its sister topic of ergodic theory can also be emphasized through symbolization by using the language inherent in information theory. Information as described by the Shannon information theory begins with questions regarding code length necessary for a particular message. Whether the message be a poem by Shakespeare, a raster scanned or a multiscaled (wavelet) represented image, or even an initial condition and its trajectory describing the evolution under a given dynamical process, the language of information theory proves to be highly powerful and useful. In the first few sections of this chapter, we will review just enough classical information theory to tie together some strong connections to dynamical systems and ergodic theory in the later sections.

9.1 A Little Shannon Information on Coding by Example

Putting the punchline first, we will roughly state that information is defined to describe a decrease in uncertainty. Further, less frequent outcomes confer more information about a system. The Shannon entropy is a measure of average uncertainty.

Basic questions of data compression begin a story leading directly to Shannon entropy and information theory. For ease of presentation, we shall introduce this story in terms of representation of a simple English phrase. The discussion applies equally to phrases in other human languages, representations of images, encodings of music, computer programs, etc.

The basic idea behind *entropy coding* is the following simple principle:

- We assign short code words to likely frequent source symbols and long code words to rare source symbols.
- Such source codes will therefore tend to be variable length.
- Since long code words will be assigned to those sources which are less likely, they are therefore more surprising, and conversely short codes are less surprising.

Consider a phrase such as the single English word "Chaos"; we choose a single-word phrase only to make our presentation brief. This story would be the same if we were to choose a whole book of words, such as this book in your hands. Encoded in standard ASCII,[116]

$$\text{Chaos} \to 1000011\ 1101000\ 1100001\ 1101111\ 1110011. \tag{9.1}$$

For convenience and clarity to the reader, a space is indicated between each 7-bit block to denote individual letters. Notice that in this 7-bit version of standard ASCII coding, it takes $5 \times 7 = 35$ bits to encode the 5 letters in the word Chaos, stated including the uppercase beginning letter,

$$\text{"C"} \to 1000011, \tag{9.2}$$

versus what would be the lowercase in ASCII,

$$\text{"c"} \to 1100011. \tag{9.3}$$

ASCII is a useful code in that it is used universally on computers around the world. If a phrase is encoded in ASCII, then both the coder and the decoder at the other end will understand how to translate back to standard English, using a standard ASCII table. The problem with ASCII, however, is that it is not very efficient. Consider that in ASCII encoding

$$\text{"a"} \to 1111010, \quad \text{"z"} \to 1100001. \tag{9.4}$$

Both the "a" and the "z" have reserved the same 7-bit allocation of space in the encoding, which was designed specifically for English. If it were designed for some language where the "a" and the "z" occurred equally frequently, this would be fine, but in English, it would be better if the more frequently used letters, such as vowels, could be encoded with 1- or 2-bit words, say, and those which are rarely used, like the "z" (or even more so the specialty symbols such as \$, &, etc.), might be reasonably encoded with many bits. On average such an encoding would do well when used for the English language for which it was designed.

Codes designed for specific information streams, or with *assumed prior* knowledge regarding the information streams can be quite efficient. Amazingly, encoding efficiencies of better than 1-bit/letter may even be feasible. Consider the (nonsense) phrase with 20 characters,

$$\text{"chchocpohccchohhchco"}. \tag{9.5}$$

In ASCII it would take $20 \times 7 = 140$ bits for a bit rate of 7 bits/letter. However, in a Huffman code[117] we can do much better. Huffman coding requires a statistical model regarding the expected occurrence rate of each letter. We will take as our model[118]

$$p_1 = P(\text{"c"}) = 0.4,\ p_2 = P(\text{"h"}) = 0.35,\ p_3 = P(\text{"o"}) = 0.2,\ p_4 = P(\text{"p"}) = 0.05, \tag{9.6}$$

[116] American Standard Code for Information Interchange (ASCII) is a character encoding of the English alphabet used commonly in computers. Each character gets the same length of 7 bits even though that some characters are not likely to be used.

[117] The Huffman code is a variable-length code algorithm that is in some sense optimal, as discussed in Section 9.2, especially in Theorem 9.1. This breakthrough was developed by an MIT student, D.A. Huffman, and published in his 1952 paper [167].

[118] The notation $P(A)$ denotes the probability of event A.

which we derive by simply counting occurrences[119] of each letter to be 8, 7, 4, and 1, respectively, and by assuming stationarity.[120] With these probabilities, it is possible the following Huffman code follows,

$$\text{“c”} \to 0, \quad \text{“h”} \to 10, \quad \text{“o”} \to 110, \text{“p”} \to 111, \tag{9.7}$$

from which follows the Huffman encoding,

$$\text{“chchocpohcccchohhchc”} \to$$

$$0\ \ 10\ \ 0\ \ 10\ \ 110\ \ 0\ \ 111\ \ 110\ \ 10\ \ 0\ \ 0\ \ 0\ \ 10\ \ 110\ \ 10\ \ 10\ \ 0\ \ 10\ \ 0\ \ 110. \tag{9.8}$$

Again, spaces are used here to guide the eye separating bits related to each individual letter. The spaces are not actually part of the code. In this encoding, the 20-letter phrase chchocpohcccchohhchc is encoded in 37 bits, for a bit rate of 37 bits/20 letters = 1.85 bits/letter, which is a great deal better than what would have been 7 bits/letter in a 140-bit ASCII encoding.

We have not given the details behind how to form a Huffman code from a given discrete probability distribution as this would be somewhat outside the scope of this book and beyond our needs here. Our point is simply that there are much better ways to encode an information stream by a well-chosen variable-length code as exemplified by the well regarded Huffman code. The dramatic improvement of course comes from this prettygood statistical model used. For example, zero bit resource is allocated for the letter “z”, and the rest of the alphabet. Any message that requires those symbols would require a different encoding or the encoding simply is not possible.

How shall the quality of a coding scheme such as the Huffman code be graded? Considering the efficiency of the encoding is a great deal like playing the role of a bookie,[121] hedging bets. Most of the bit resource is allocated for the letter “c” since it is most common, and the least is allocated for the “o” and “p” since they are less common than the others used. When a message is stored or transmitted in agreement with this statistical model, then high efficiency occurs and low bit rates are possible. When a message that is contrary to the model is transmitted with this then ill-fitted model, less efficient coding occurs. Consider a phrase hhhhhhhhhhhhhhhhhhhh (“h” 20 times).[122] Then a 3 bits/letter efficiency occurs, which is a worst-case scenario with this particular Huffman coding. That is still better than ASCII, because the model still assumes only 4 different letters might occur. In other words, it still beats ASCII since at the outset we assumed that there is zero probability for all but those 4 symbols.

[119] Without speaking to the quality of the model, note that counting occurrences is surely the simplest way to form a probabilistic model of the likelihood of character occurrences.

[120] Roughly stated, a stochastic process is stationary if the joint probabilities “do not change in time.” More will be said precisely in Definition 9.12 in Section 9.5.

[121] A bookie is a person who handles bets and wagers on events, usually sporting events on which gamblers place money in hopes that their favorite team will win and they will win money. Bookies need a good probability model if they expect to win over many bets on average.

[122] Consider another coding method entirely, called run length encoding, in which the “h” is to be repeated some number of times [262]. The longer the repetition, the more efficient such an encoding would be, since the overhead of the annotations to repeat is amortized; the codings of “state 20 h’s” and “state 50 h’s” are essentially the same, for example. Such codes can only be useful for very special and perhaps trivial phrases with long runs of single characters. This discussion relates to the notion of **Kolmogorov complexity,** which is defined to be the length of the optimal algorithm which reproduces the data. Kolmogorov complexity is not generally computable.

To make the notion of bit rate and efficiency more rigorous, note that the statistical *expectation* of the bit rate in units of bits/letter may be written

$$\begin{aligned}\text{Average bit rate} &= \text{bits/letter}\\ &= \sum_i P(i\text{th letter occurs in message}) \text{ (Length used to encode } i\text{th letter)}\\ &= 0.4*1+0.35*2+0.2*3+0.05*3\text{bits/letter}\\ &= 1.85\text{bits/letter.}\end{aligned} \tag{9.9}$$

The perfect coincidence in this example between expectation and actual encoding rate simply reflects the fact that the toy message used matches the probabilities. The optimal bit length of each encoding can be shown to be bounded,

$$(\text{Length used to encode } i\text{th letter}) \leq -\log_2(p_i). \tag{9.10}$$

Consider the relation of this statement to the optimal code rate implicit in Theorem 9.2.

Considering an optimal encoding of a bit stream leads to what is called the Shannon entropy, defined formally in Definition 9.4, specialized for a coding with 2 outcomes (bits):

$$H_2 :\equiv -\sum_i p_i \log_2(p_i). \tag{9.11}$$

Shannon entropy carries the units of bits/letter, or alternatively bits/time if the letters are read at a rate of letters/time. Comparing this to the question of how long the coding is in the previous example, Eq. (9.6),

$$H_2 = -0.4\log_2(0.4) - 0.35\log_2(0.35) - 0.2\log_2(0.2) - 0.05\log_2(0.05) = 1.7394. \tag{9.12}$$

Note that $1.7394 < 1.85$ as the code used was not optimal. The degree to which it was not optimal is the degree to which, in Eq. (9.10),

$$-[(\text{Length used to encode } i\text{th letter}) + \log_2(p_i)] > 0. \tag{9.13}$$

Specifically, notice that $p_3 = 0.2 > p_4 = 0.05$, but they each are "gambled" by the bit rate bookie who allocates 3 bits each. There are reasons for the suboptimality in this example:

- The probabilities are not d-atic (d-atic with $d = 2$, also spelled dyatic in this case, means $p_i = \frac{r}{2^n}$ for some r, n integers and for every i).
- A longer version of a Huffman coding would be required to differentiate these probabilities. Here only three bits were allowed at maximum. Huffman codes are developed in a tree, and depth is important too.
- Furthermore, the Huffman coding is a nonoverlapping code [167], meaning each letter is encoded before the next letter can be encoded. Huffman code is a special example of so-called entropy coding within the problem of lossless[123] compression.

[123] The object of lossless encoding is to be able to recover the original "message" exactly, as opposed to lossy encoding, in which some distortion is allowed. For example, consider a computer program as the source. A "zipped" file of a computer program must be decompressed *exactly* as the original for the decompressed computer program to work. On the other hand, lossless compression generally borrows from representation theory of functions; a lossy scheme for compressing a digital photograph includes truncating a Fourier expansion.

Huffman codes are optimal within the class of entropy coding methods, beating the original Shannon–Fano code, for example, but it is not as efficient as the Lempel–Ziv (LZ) [324] or arithmetic coding methods [195, 68]. Our purpose in using the Huffman coding here was simply a matter of specificity and simplicity of presentation.

Important aspects of a proper coding scheme are that it must at least be

- One-to-one and therefore invertible[124]: Said otherwise, for each coded word, there must be a way to recover the original letters so as to rebuild the word. Without this requirement, we could quite simply compress every message no matter how long, say the complete works of Shakespeare, to the single bit 0, and not worry that you cannot go back based on that bit alone. The information is lost as such.
- Efficient: A poor coding scheme could make the message length longer than it may have been in the original letter coding. This is a legal feature of entropy encoding, but of course not useful.

The details of these several coding schemes are beyond our scope here, but simply knowing of their existence as related to the general notions of coding theory is leading us to the strong connections of entropy in dynamical systems. Sharpening these statements mathematically a bit further will allow us to discuss the connection.

9.2 A Little More Shannon Information on Coding

The Shannon entropy $H_D(X)$ defined in Definition 9.4 can be discussed in relation to the question of the possibility of an optimal code, as the example leading to Eq. (9.9) in the previous section reveals. To this end, we require the following notation and definitions.

Definition 9.1. An **encoding** $c(x)$ for a random variable X (see Definition 3.3) is a function from the countable set $\{x\}$ of outcomes of the random variable to a string of symbols from a finite alphabet, called a D-ary code.[125]

Remark 9.1. Commonly in digital computer applications which are based on binary bits, representing "0" and "1", or "on" and "off", generally $D = 2$.

Following the above discussion, it is easy to summarize with the following definition.

Definition 9.2. The expectation of the length L of a source encoding $c(x)$ of the random variable X with an associated probability distribution function $p(x)$ is given by

$$L(C) = \sum_x p(x) l(c(x)), \tag{9.14}$$

where $l(c(x))$ is the length of the encoding $c(x)$ in units of bits. In Eq. (9.9), we described the units of L to be bits/letter; however, it can be interpreted also simply as length when a fixed positive number C of letters are coded.

[124] A code is called **nonsingular** if for every outcome there is a unique representation by a string from the symbol set ($\{0,1\}$ if binary); otherwise it is called singular.

[125] D-ary refers to the use of D symbols, which may be taken from the symbol set, $\{0,1,\ldots,D-1\}$ or $\{0,1\}$, the usual binary set if $D = 2$. Arithmetic occurs in base-D.

Given an encoding and repeated experiments from a random variable, we can get a code extension which is simply an appending of codes of each individual outcome.

Definition 9.3. Given a code $c(x)$, an **encoding extension** is a mapping from ordered strings of outcomes x_i to an ordered string of symbols from the D-ary alphabet of the code,

$$C = c(x_1 x_2 \dots x_n) \equiv c(x_1)c(x_2)\dots c(x_n). \tag{9.15}$$

This is a concatenation of the alphabet representation of each outcome in the sequence $x_1, x_2, \dots, x_n$.

The formal definition of Shannon entropy can be stated as follows.

Definition 9.4. The **Shannon entropy** of a D-ary code for a random X with probability distribution function $p(x)$ is given by the nonnegative function

$$H_D(X) = -\sum_x p(x) \log_D p(x), \tag{9.16}$$

in terms of the base-D logarithm.

There is a strong connection between the notion of optimal coding and this definition of entropy as revealed by the following classic theorems from information theory, as proven in [68]. Discussion of the existence of optimal codes follows starting from the Kraft inequality.

Theorem 9.1. An instantaneous code[126] C of a random variable X with code word lengths $l(x)$ satisfies the inequality

$$\sum_x D^{-l(x)} \leq 1. \tag{9.17}$$

By converse, the Kraft inequality implies that such an instantaneous code exists. The proof of this second statement is by Lagrange multiplier optimization methods [68]. Furthermore a statement relating Shannon entropy and expected code length $L(C)$ can be summarized as follows.

Theorem 9.2. Given a code C of a random variable X with probability distribution $p(x)$, C is a minimal code if the code word lengths are given by

$$l(x) = -\log_D p(x). \tag{9.18}$$

The following definition of Shannon information may be described as a pointwise entropy in that it describes not only the length of a given optimally coded word, but also the entropy as if we know that the random variable takes on $X = x$, and therefore the corresponding code word is used.

[126] An **instantaneous code**, also called a **prefix code**, completes each code word before the next word begins. No code word is a prefix of any other code word. Therefore, the receiver does not require a prefix before each word to know when to start reading the next word. The converse is that two words share the same coding, and therefore a prefix would be required to distinguish them. Since perhaps the most commonly used prefix code is the Huffman code, favored for its optimality properties, often by habit even other source codes are called Huffman by some.

Definition 9.5. Shannon information is defined by the quantity

$$l(x) = -\log_D p(x). \tag{9.19}$$

Shannon information in some sense describes *a degree of surprise we should hold when an unlikely event comes to pass*. A great deal more information is inferred when the unlikely occurs than when the usual, high probability outcomes x occur. Comparing Eqs. (9.14), (9.16), and (9.19), we can describe Shannon entropy $H_D(X)$ as an *information expectation* of the random variable.

Now we can state the relationship of an optimal code C^* to the entropy of the process, which gives further meaning to the notion of Shannon entropy.

Theorem 9.3 (source coding). Let C^* be an optimal code of a random variable X, meaning the expected code length of any other code C is bounded, $L(C) \geq L(C^*)$, and if C^* is an instantaneous D-ary code, then

$$H_D(X) \leq L(C^*) \leq H_D(X) + 1. \tag{9.20}$$

Following the example in the previous section, recall that in the ASCII coded version of the 20-character message "chchocpohccchohhchc" in Eq. (9.8), $L(C) = 140$, whereas with the Huffman coded version, $L(C^*) = 37$ bits. Furthermore, with the statement that Huffman is optimal, we know that this is a shortest possible encoding by an instantaneous code. Since Huffman coding embodies an algorithm which provides optimality, this validates existence.

Theorem 9.4. A Huffman code is an optimal instantaneous code.

That Huffman is an optimal code is enough for our discussion here regarding existence of such codes, and the relationship between coding and entropy. The algorithmic details are not necessary for our purposes in this book, and so we skip the details for the sake of brevity. Details on this and other codes, Lempel–Ziv (LZ) [324] or arithmetic coding methods [195, 68] most notably, can be found elsewhere. As it turns out, other nonprefix codes, notably arithmetic coding, can yield even shorter codes, which is a play on the definitions of optimal, code, and encoding extension. In brief, arithmetic coding can be understood as a mapping from letters to base-D representations of real numbers in the unit interval $[0,1]$, distributed according to the probabilities of X, and perhaps in levels by a stochastic process $X_1, X_2, \ldots$ generating letters. In this perspective, a Huffman code is a special case encoding one letter at a time by the same mapping process, whereas arithmetic coding maps the entire message all together. Thus arithmetic coding allows the reading of letters in seemingly overlapping fashion.

9.3 Many Random Variables and Taxonomy of the Entropy Zoo

Given many random variables,

$$\{X_1, X_2, \ldots, X_n\} \tag{9.21}$$

(or two when $n = 2$) in a product probabilities space,

$$\{(\Omega_1, \mathcal{A}_1, P_1) \times (\Omega_2, \mathcal{A}_2, P_2) \times \cdots \times (\Omega_n, \mathcal{A}_n, P_n)\}, \tag{9.22}$$

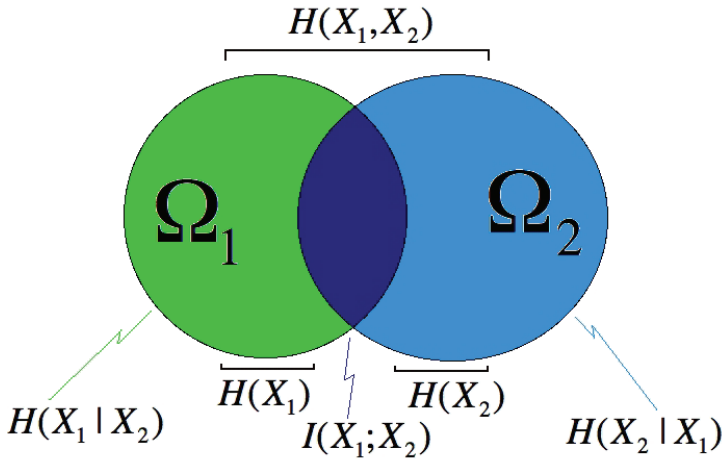

Figure 9.1. *Given a compound event process, here $n = 2$ with Ω_1 and Ω_2 associated with Eqs.* (9.21) *and* (9.22)*, we can discuss the various joint, conditional, and individual probabilities, as well as the related entropies as each is shown with their associated outcomes in the Venn diagram.*

one can form probability spaces associated with the many different intersections and unions of outcomes; see Fig. 9.1. Likewise, the associate entropies we will review here give the degree of "averaged surprise" one may infer from such compound events.

Definition 9.6. The **joint entropy** associated with random variables $\{X_1, X_2, \ldots, X_n\}$ is

$$H(X_1, X_2, \ldots, X_n) = -\sum_{x_1, x_2, \ldots, x_n} p(x_1, x_2, \ldots, x_n) \log p(x_1, x_2, \ldots, x_n) \tag{9.23}$$

in terms of the joint probability density function $p(x_1, x_2, \ldots, x_n)$, and with the sum taken over all possible joint outcomes $(x_1, x_2, \ldots, x_n)$.

Joint entropy is sometimes called the *total entropy* of the combined system. See Fig. 9.1, where $H(X_1, X_2)$ is presented as the uncertainty of the total colored regions.

Definition 9.7. The **conditional entropy** associated with two random variables $\{X_1, X_2\}$ is

$$H(X_1|X_2) = -\sum_{x_2} p(x_2) H(X_1|X_2 = x_2) \tag{9.24}$$

in terms of the probability density function $p_2(x_2)$.

Conditional entropy $H(X_1|X_2)$ can be understood as the *remaining entropy* bits in the uncertainty of the random variable X_1 with the information bits already given regarding the intersection events associated with X_2. See the Venn diagram in Fig. 9.1. In other words, measuring $H(X_1|X_2)$ answers the question, "What does X_2 not say about X_1?" An alternative formula for conditional entropy may be derived in terms of the joint probabilities $p(x_1, p_2)$,

$$H(X_1|X_2) = -\sum_{(x_1, x_2)} p(x_1, x_2) \log \frac{p(x_1, x_2)}{p(x_2)}, \tag{9.25}$$

which is easy to see since the term $H(X_1|X_2 = x_2)$ given in Definition 9.7:

$$H(X_1|X_2 = x_2) = \sum_{x_1} p(x_1|x_2) \log p(x_1|x_2) \tag{9.26}$$

Using the relationship for conditional probabilities,

$$p(x_1|x_2) = \frac{p(x_1,x_2)}{p_2(x_2)}, \tag{9.27}$$

substitution into Definition 9.7 yields

$$H(X_1|X_2) = -\sum_{x_2} p_2(x_2) H(X_1|X_2 = x_2) \tag{9.28}$$

$$= -\sum_{x_1,x_2} p_2(x_2) p(x_1|x_2) \log(x_1|x_2) \tag{9.29}$$

$$= -\sum p(x_1,x_2) \log p(x_1|x_2), \tag{9.30}$$

the last being a statement of a cross entropy. Finally, again applying Eq. (9.27),

$$-\sum p(x_1,x_2) \log p(x_1|x_2) = -\sum p(x_1,x_2) \log \frac{p(x_1,x_2)}{p_2(x_2)} \tag{9.31}$$

$$= -\sum p(x_1,x_2) \log p(x_1,x_2) \tag{9.32}$$

$$+\sum p(x_1,x_2) \log p_2(x_2).^{127} \tag{9.33}$$

From this follows the **chain rule** of entropies:

$$H(X_1|X_2) + H(X_2) = H(X_1,X_2). \tag{9.34}$$

A few immediate statements regarding the relationships between these entropies can be made.

Theorem 9.5. $H(X_1|X_2) = 0$ if and only if X_1 is a (deterministic) function of the random variable X_2.

In other words, since X_1 can be determined whenever X_2 is known, the status of X_1 is certain for any given X_2.

Theorem 9.6. $H(X_1|X_2) = H(X_1)$ if and only if X_1 and X_2 are independent random variables.[128]

This can be understood as a statement that knowing X_2 gives no further information regarding X_1 when the two random variables are independent.

Two useful further entropy-like measures comparing uncertainty between random variables are the mutual information and the Kullback–Leibler divergence.

[127] **cross entropy KL

[128] X_1 and X_2 are defined to be independent if $p(x_1,x_2) = p_1(x_1)p_2(x_2)$, or likewise by Eq. (9.27), $p(x_1|x_2) = p_1(x_1)$ and $p(x_2|x_1) = p_2(x_2)$.

Definition 9.8. The **mutual information** associated with two random variables $\{X_1, X_2\}$ is

$$I(X_1; X_2) = \sum_{x_1, x_2} p(x_1, x_2) \log \frac{p(x_1, x_2)}{p_1(x_1)\, p_2(x_2)}.^{129} \tag{9.35}$$

Alternatively, there follows another useful form of the same,

$$I(X_1; X_2) = H(X_1) - H(X_1|X_2). \tag{9.36}$$

Mutual information may be understood as the amount of information that knowing the values of either X_1 or X_2 provides about the other random variable. Stated this way, mutual information should be symmetric, and it is immediate to check that Eq. (9.35) is indeed so. Likewise, inspecting the intersection labeled $I(X_1; X_2)$ in Fig. 9.1 also suggests the symmetric nature of the concept. An example application to the spatiotemporal system pertaining to global climate from [104] is reviewed in Section 9.9.2.

The Kullback–Leibler divergence on the other hand is a distance-like measure between two random variables, which is decidedly asymmetric.

Definition 9.9. The **Kullback–Leibler divergence** between the probability density functions p_1 and p_2 associated with two random variables X_1 and X_2 is

$$D_{KL}(p_1 || p_2) = \sum_x p_1(x) \log \frac{p_1(x)}{p_2(x)}. \tag{9.37}$$

The D_{KL} is often described as if it is a metric-like distance between two density functions, but it is not technically a metric since it is not necessarily symmetric; generally,

$$D_{KL}(p_1 || p_2) \neq D_{KL}(p_2 || p_1). \tag{9.38}$$

Nonetheless, it is always nonnegative, as can be seen from (9.2) by considering $-\log(p_2(x))$ as a length of encoding. Furthermore, $D_{KL}(p_1 || p_2)$ can be understood as an entropy-like measure in that it measures the expected number of extra bits which would be required to code samples of X_1 when using the wrong code as designed based on X_2, instead of purpose designed for X_1. This interpretation can be understood by writing

$$\sum_x p_1(x) \log \frac{p_1(x)}{p_2(x)} = \sum_x p_1(x) \log p_1(x) - \sum_x p_1(x) \log p_2(x) \tag{9.39}$$

$$= H_c(X_1, X_2) - H(X_1), \tag{9.40}$$

where $H_c(X_1, X_2)$ is the cross entropy,

Definition 9.10. The **cross entropy** associated with two random variables $\{X_1, X_2\}$ with probability density functions p_1 and p_2,

$$H_c(X_1|X_2) = H(X_1) + D_{KL}(p_1 || p_2), \tag{9.41}$$

describes the inefficiency of using the wrong model p_2 to build a code for X_1 relative to a correct model p_1 to build an optimal code whose efficiency would be $H(X_1)$.

[129] It is useful to point out at this stage that $p_1(x_1)$ and $p_2(x_2)$ are the **marginal distributions** of $p(x_1, x_2)$; $p_1(x_1) = \sum_{x_2} p(x_1, x_2)$ and, likewise, $p_2(x_2) = \sum_{x_1} p(x_1, x_2)$.

Thus when $p_1 = p_2$ and, therefore, $D_{KL}(p_1||p_2) = 0$, the coding inefficiency as measured by cross entropy $H_c(X_1|X_2)$ becomes zero. Mutual information can then be written:

$$I(X_1;X_2) = D_{KL}(p(x_1,x_2)||p_1(x_1)p(x_2)). \tag{9.42}$$

A stochastic process allows consideration of entropy rate.

Definition 9.11. Given a stochastic process $\{X_1,X_2,\ldots\}$, the **entropy rate** is defined in terms of a limit of joint entropies,

$$H = \lim_{n\to\infty} \frac{H(X_1,X_2,\ldots,X_n)}{n}, \tag{9.43}$$

if this limit exists.

Assuming the special case that X_i are independent and identically distributed (i.i.d.), and noting that independence[130] gives

$$p(x_1,x_2,\ldots,x_n) = \prod_{i=1}^{n} p(x_i), \tag{9.44}$$

it follows quickly that[131]

$$H(X_1,X_2,\ldots,X_n) = \sum_{i=1}^{n} H(X_i) = nH(X_1). \tag{9.45}$$

The second statement in this equality only requires independence, and the third follows the identically distributed assumption. Now we are in a position to restate the result (9.20) of the source coding Theorem 9.3 in the case of a coding extension. If $L(c(x_1x_2\ldots x_n))$ is the length of the coding extension of n coded words of realizations of the random variables $X_1,X_2,\ldots,X_n$, then Eq. (9.20) generalizes to a statement regarding the minimum code word *length expected per symbol*,

$$H(X_1,X_2,\ldots,X_n) \le nL(C) \le H(X_1,X_2,\ldots,X_n)+1, \tag{9.46}$$

or, in the case of a stationary process,

$$\lim_{n\to\infty} L = H, \tag{9.47}$$

the entropy rate. Notice that this length per symbol is just what was emphasized by example near Eq. (9.9). In the case of an i.i.d. stochastic process, Eq. (9.46) specializes to

$$H(X_1) \le L(C) \le H(X_1)+\frac{1}{n}. \tag{9.48}$$

[130] Statistical independence of X_1 from X_2 is defined when given probabilities as follows: $p(x_1,x_2) = p(x_1)$ for each $X_1 = x_1$ and $X_2 = x_2$. And since $p(x_1,x_2) = p(x_1|x_2)p(x_2)$, then independence implies $p(x_1,x_2) = p(x_1)p(x_2)$. Likewise, $P(x_2|x_1) = p(x_2)$ since $p(x_2|x_1) = \frac{p(x_1,x_2)}{p(x_1)} = \frac{p(x_1)p(x_2)}{p(x_1)} = p(x_2)$.

[131] $H(X_1,X_2,\ldots,X_n) = \sum_i p(x_1,x_2,\ldots,x_n)\log p(x_1,x_2,\ldots,x_n) = \sum_i \prod_{i=1}^n p(x_i)\log\prod_{i=1}^n p(x_i) = \sum_i \prod_{i=1}^n p(x_i)[\sum_{i=1}^n \log p(x_i)] = \sum_i \prod_{i=1}^n p(x_i)[\sum_{i=1}^n \log p(x_i)]$.

This expression is symbolically similar to Eq. (9.20), but now we may interpret the entropy rate of the i.i.d. stochastic process to the expected code word length of a coded extension from appending many coded words.

Finally in this section, coming back to our main purpose here to relate information theory to dynamical systems, it will be useful to introduce the notion of channel capacity. We recall that according to the channel coding theorem, which we will discuss below, a chaotic oscillator can be described as such a channel. Channel capacity is the answer to the question, "How much information can be transmitted or processed in a given time?" One may intuit that a communication system may degrade in the sense of increasing error rate as the transmission rate increases. However, this is not at all the case, as the true answer is more nuanced. The part of the channel coding theorem that we are interested in here can be stated as follows.

Theorem 9.7 (channel coding theorem). A "transmission" rate R is achievable with vanishingly small probability if $R < C$, where C is the **information channel capacity**,

$$C = \max_{p(x)} I(X;Y), \tag{9.49}$$

where the maximum is taken over all possible distributions $p(x)$ on the input process, and Y is the output process.

Now to interpret this theorem, a given communication system has a maximum rate of information C known as the channel capacity. If the information rate R is less than C, then one can approach arbitrarily small error probabilities by careful coding techniques, meaning cleverly designed codes, even in the presence of noise. Said alternatively, low error probabilities may require the encoder to work on long data blocks to encode the signal. This results in longer transmission times and higher computational requirements. The usual transmission system as a box diagram is shown in Fig. 9.2, including some description in the caption of standard interpretations of the inputs and outputs X and Y. Fano's converse about error rate relates a lower bound on the error probability of a decoder,

$$H(X|Y) \leq H(e) + p(e)\log(r), \tag{9.50}$$

where

$$H(X|Y) = -\sum_{i,j} p(x_i, y_i)\log p(x_i|y_j) \quad \text{and} \quad p(e) = \sup_i \sum_{i \neq j} p(y_j|x_i). \tag{9.51}$$

To interpret the relevance of the Theorem 9.7 in the context of dynamical systems theory requires only the work of properly casting the roles of X and Y, as is discussed in Section 9.4 and illustrated in Fig. 9.6.

9.4 Information Theory in Dynamical Systems

In Chapter 6 we highlighted the description of a dynamical system as an underlying symbolic dynamics. Here we will further cement this connection by describing a complementary information theoretic perspective of the symbolic dynamics.

Perhaps there is no better way of emphasizing the information theoretic aspects of chaotic dynamical systems with an underlying symbolic dynamics than by explicitly

Figure 9.2. *Channel capacity* (9.52) *box diagram. Random variable X describes the random variable descriptive of the possibilities of the input states message. Coming out of the channel is the output states whose values are according to the random variable Y which one wishes to be closely related (i.e., desires to be identical in many applications). Alternatively, let Y be understood as states of a decoder. Such will be the case if transmission rate is slow enough, $R < C$, according to Theorem* 9.7.

.

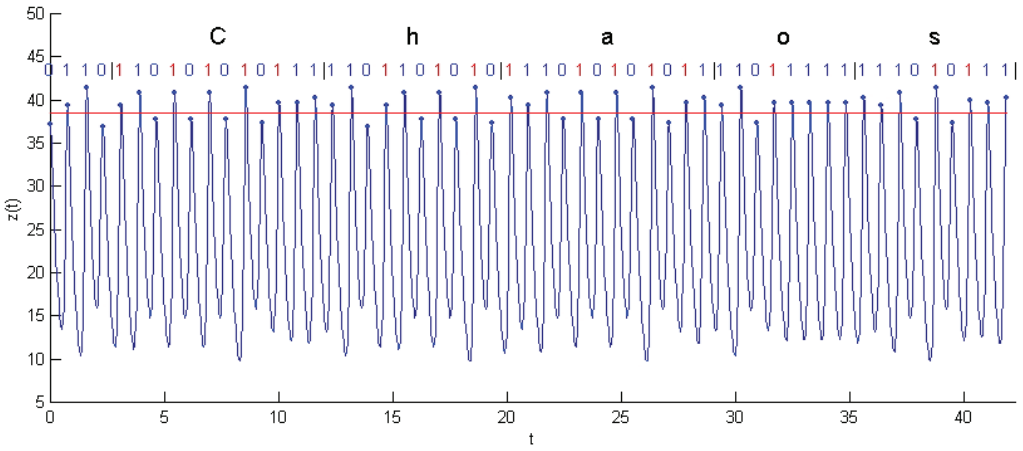

Figure 9.3. *The Lorenz attractor has chaotic orbits and an underlying symbolic dynamics, usually presented according to the successive maxima map of the $z(t)$ time series, which was already discussed in Eq.* (6.4) *according to Eqs.* (6.5)–(6.6)*, and the resulting map was already illustrated in Figs.* 6.3 *and* 6.4*. In this case, the property of chaos that there are infinitely many periodic orbits has been leveraged against the symbolic dynamics to choose periodic orbits whose symbolic dynamics encode the word "Chaos" in ASCII form, subject to the non-information-bearing bits shown in red. Compare to the embedding form of the attractor shown in Fig.* 9.4.

demonstrating orbits bearing messages as we wish. In Fig. 6.2, a time series of an orbit segment from a Lorenz equation (6.4) flow is plotted along with its phase space representation of the attractor, in Fig. 6.3. We repeat similar Figs. 9.3 and 9.4. Again, we can read symbols from the $z(t)$ time series by the symbol partition (6.6); zeros and ones can be read by the position of the local maxima of the $z(t)$ coordinate from the differential equation relative to the cusp-maximum of this value z_n in a one-dimensional ansatz $z_{n+1} = f(z_n)$. A 0 bit in the message as indicated in the time series in Fig. 9.4 corresponds to a relatively smaller local maximum (corresponding to the left side of the cusp in Fig. 9.4), meaning local maximum (a red point) occurs on the left "wing" of the attractor. Likewise, 1 bits are encoded in the orbit. Now, however, we choose to read the symbols and interpret them as if they are coded words in ASCII.

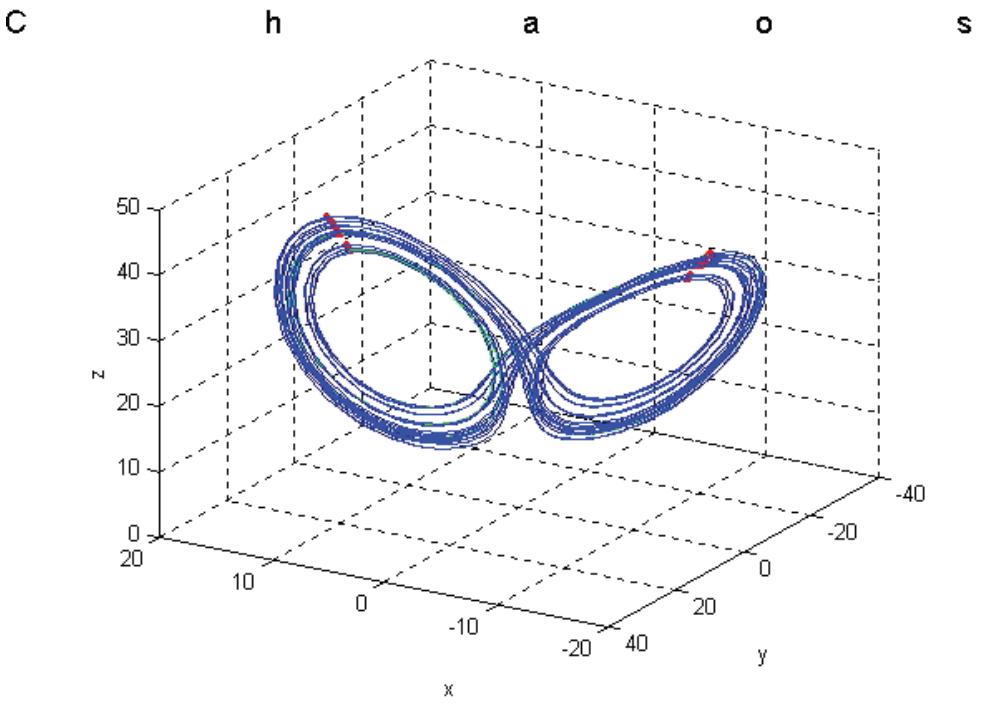

Figure 9.4. *The time series shown in Fig.* 9.3 *of an orbit segment spelling "Chaos" shown in its phase space presentation of* (x,y,z) *coordinates on the chaotic attractor. This is a carefully chosen initial condition.*

With the Lorenz parameters set to be the usual famous values as they were in the previous chapter, (10,28,8/3), we may observe that the underlying symbolic dynamics grammar never allows two zeros in a row in a symbolic word string, as is described in Fig. 6.26. Therefore, a suitable source code can be built on a prefix coding of the transitions in the graph shown in Fig. 9.5 (right). There are two possible outcomes from a previous symbol 1; either a 0 or a 1 can follow. However, there is only one outcome possible from a symbol 0; simply stated, there is no surprise from the transition, by observing the 1 bit which follows a 0 bit—it was the only possibility. So this transition can be said to be non-information-bearing, or a zero entropy state. To emphasize this, we labeled this transition * in the directed graph since it serves only a role as a pause required to transition back to an information-bearing state—that is, a state where either a 0 *or* a 1 might follow.

Specifically, in terms of this source coding and using the ASCII prefix code, the word "Chaos" has been encoded into the chaotic oscillations. All the non-information-bearing 1/s denoting the $*$-transition in the directed graph in Fig. 9.5 are those 1's which are colored red. Counting the ASCII code length to carry the word "Chaos" is $7 \times 5 = 35$ bits, but including the non-information-bearing bits has further required $16 + 35 = 51$ bits. These extra bits represent a nonmaximal channel capacity of the message carrier, which in this case is the dynamical system itself.

The dynamical system encoding the word "Chaos," or any other word, phrase, or arbitrary information stream including sounds or images, etc. [158, 26, 251, 55], has a fundamental associated channel capacity C, (9.52). Thus, by Theorem 9.7, transmission

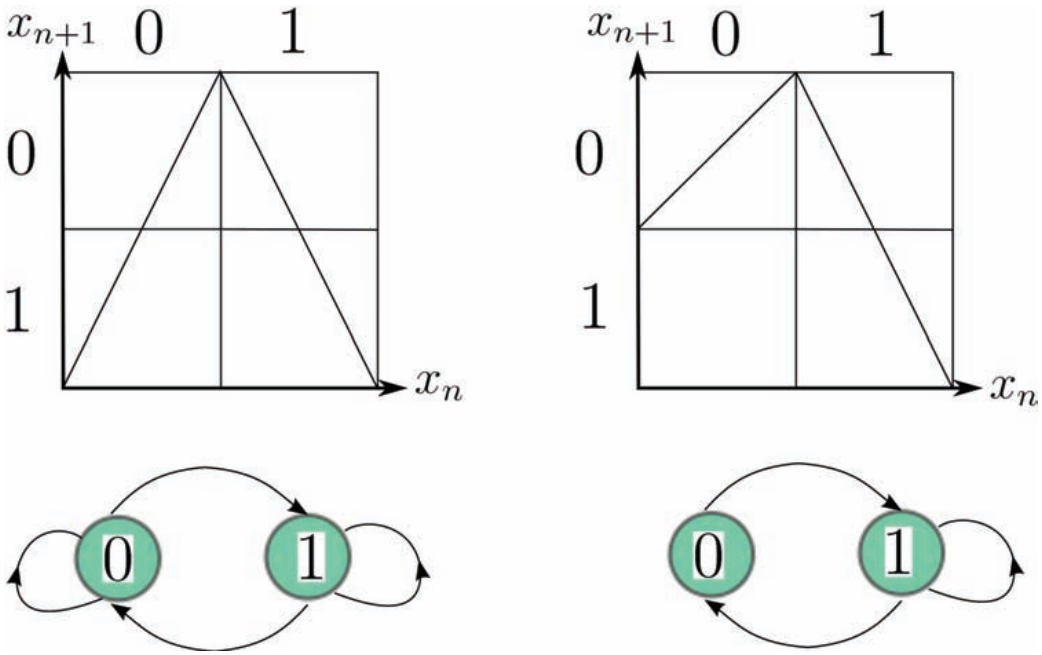

Figure 9.5. *Transitions with no surpise carry no information. (Left) This example full tent map on the Markov partition shown has orbits that at all possible times n, $x_n \in [0,1]$, there exist nearby initial conditions whose evolution will quickly evolve to a state with an opposite symbol* 0 *or* 1 *as the case may be; this unrestricted symbolic grammar is generated by the simple (dyadic) two-state directed graph shown below. Each state allows transition to a* 0 *or* 1 *outcome, so the surprise of observing the outcome is the information born by random walks through the graph corresponding to iterating the map. (Right) This piecewise linear map drawn on its Markov transition has a symbolic dynamics generated by the graph shown below. This grammar does not allow a* 0 *symbol to follow a* 0 *symbol. Thus when at the* 0 *labeled state ($x < c$), only a* 1 *can possibly follow; this transition bears no surprise. Said equivalently, it is not information bearing. Thus the required* 1 *transition serves as nothing other than a delay or pause in the transmission of a message to an information-bearing state—the grammar allows no two* 0*s in a row. From* 1 *either a* 0 *or a* 1 *may follow. Compare to Fig.* 9.3*, where such a grammar is used to encode oscillations in a Lorenz attractor to transmit a real message.*

rates R less than C are achievable. This imposes a rate of those non-information-bearing bits we depicted as *red* in Fig. 9.3. See Fig. 9.6, where we illustrate the reinterpretation of the standard box diagram shown in Fig. 9.2 in the setting where the channel is taken to be the dynamical system. In such a case, we reinterpret Eq. (9.52) as

$$C = \max_{p(x)} I(X_n; X_{n+1}). \tag{9.52}$$

The entropy which is most relevant uses the so-called maximum entropy measure (Theorem 9.8) corresponding to the topological entropy $h_{top}(\Sigma')$, which in Eq. (6.74) we

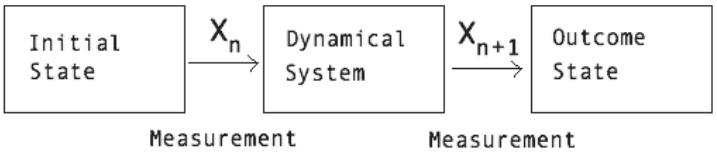

Figure 9.6. *Channel capacity* (9.52) *box diagram interpreted as a dynamical system. Compare to Fig.* 9.2 *and Theorem* 9.7.

see is descriptive of the number of allowable words N_n of a given length n. Perhaps most revealing is the spectral description (6.67) stating that $h_{top}(\Sigma'_k) = \ln \rho(A)$. The trade-off between channel capacity, transmission rates, and noise resistance was revealed in [29]. The corresponding devil's staircase graph is shown in Fig. 6.30 by a computation illustrated by Fig. 6.29 by the spectral method.

Before closing with this example, note that feedback control can be used to stabilize orbits with arbitrary symbolic dynamics [158, 26, 29, 103] starting from control of chaos methods [243, 287]. Specifically, electronic circuitry can be and has been built wherein small control actuations may be used to cause the chaotic oscillator to transmit information by small energy perturbations with simple circuitry to otherwise high powered devices. This has been the research engineering emphasis of [66, 67] toward useful radar devices.

Stated without the emphasis on practical devices, since it can be argued that there is information concerning states in all dynamical systems and a chaotic oscillator could be characterized as a system with positive entropy, then the evolution of the system through these states corresponds to an information-generating system. These systems have been called "information baths" [149].

What does this example tell us in summary?

- Realizing chaotic oscillators as information sources that *forget current measurement states and allow information bits to be inserted at a characteristic rate* as the system evolves (or is forced to evolve by feedback control) into new states summarizes the interpretation of a dynamical system as an information channel; see Fig. 9.6.

- The uncertainty in dynamical systems is in the choice of the initial condition even if the evolution rule may be deterministic.[132] The uncertainty in the symbolic outcome is described as the random variable defining the probability of states, corresponding to symbols. That is realized correspondingly in a deterministic dynamical system as the *unknown precision of state which is amplified upon iteration of an unstable system*. Even though the evolution of the states in the phase space of the dynamical system is deterministic, the exact position in phase space is practically never known exactly. We will make this idea mathematically precise in Section 9.5.

- Small feedback control can be used to steer orbits so that those orbits may bear a symbolic dynamics corresponding to desired information.

[132] This is the difference between a dynamical system (deterministic) and a random dynamical system (non-deterministic). See, for example, the stochastic process in Eq. (3.38) which nonetheless has a deterministic evolution of densities rule by the random dynamical system's Frobenius–Perron operator, (3.43).

9.5 Formally Interpreting a Deterministic Dynamical System in the Language of Information Theory

In the previous section, we illustrated by example that through symbolic dynamics, it is quite natural to think of a dynamical system with such a representation in terms of information theory. Here we will make this analogy more formal, thus describing the connection, which is some foundational aspects of ergodic theory. In this section we will show how to understand a deterministic dynamical system as an information-bearing stochastic process. Here we will describe a fundamental information theoretic quantity called Kolmogorov Sinai entropy (KS entropy), $h_{KS}(T)$, which gives a concept of information content of orbits in measurable dynamics. By contrast, in the next section we will discuss in greater detail the topological entropy, $h_{top}(T)$.

Assume a dynamical system,

$$T : M \to M, \tag{9.53}$$

on a manifold M, with an invariant measure μ. For sake of simplicity of presentation, we will assume a symbol space of just two symbols,[133]

$$\Omega = \{0,1\}. \tag{9.54}$$

As discussed earlier,

$$\begin{aligned} s &: M \to \Omega \\ s(x) &= \chi_{A_0}(x) + \chi_{A_1}(x), \end{aligned} \tag{9.55}$$

but with an arbitrary open topological cover,

$$\overline{A_0 \cup A_1} = M, \text{ but } A_0 \cap A_1 = \emptyset, \tag{9.56}$$

and $\chi_A : M \to [0,1]$ is the indicator function on sets $A \subset M$ as usual. We further assume probability measure using μ and a corresponding random variable (Definition 3.3),

$$X : \Omega \to \mathbb{R}, \tag{9.57}$$

for any randomly chosen initial condition $x \in M$ and therefore random symbol $s(x)$. Thus with μ, let

$$p_0 = P(X = 0) = \mu(A_0), \qquad p_1 = P(X = 1) = \mu(A_1). \tag{9.58}$$

In this notation, a dynamical system describes a discrete time stochastic process (Definition 4.15) by the sequence of random variables as follows

$$X_k(\omega) = X(s(T^k(x))). \tag{9.59}$$

Now using natural invariant measure μ when it exists, we may write

$$P(X_k = \sigma) = \mu(A_\sigma), \tag{9.60}$$

[133] The symbol partition need not be generating, in which case the resulting symbol dynamics will perhaps be a positive entropy process, but not necessarily fully descriptive of the maximal entropy of the dynamical system, as discussed in Section 6.4.6 and [40].

where

$$\sigma = 0 \text{ or } 1. \tag{9.61}$$

A stochastic process has entropy rate which we describe, a probability space $(P,\mathcal{A},\Omega)$, and associated stochastic process defined by a sequence of random variables $\{X_1(\omega),X_2(\omega),\ldots\}$. The stochastic process is defined to be stationary in terms of the joint probabilities as follows, from [51], and specialized for a discrete outcome space.

Definition 9.12. A stochastic process, $X_1,X_2,\ldots$, is stationary if for all $k>0$ the process $X_{k+1},X_{k+2},\ldots$ has the same distribution as $X_1,X_2,\ldots$. In other words, for every $B\in\mathcal{B}_\infty$,

$$\begin{aligned} &P(X_1=x_1,X_2=x_2,\ldots)\\ &=P(X_{k+1}=x_1,X_{k+2}=x_2,\ldots)\ \forall k>1, \end{aligned} \tag{9.62}$$

and for each possible experimental outcome $(x_1,x_2,\ldots)$ of the random variables.

It follows [51] that the stochastic process $X_1,X_2,\ldots$ is stationary if the stochastic process $X_2,X_3,\ldots$ has the same distribution as, $X_1,X_2,\ldots$.

Now the entropy rate of such a stochastic process by Definition 9.11 is

$$H=\lim_{n\to\infty}\frac{1}{n}H^{(n)}(\omega) \tag{9.63}$$

in terms of joint entropies

$$H^{(n)}(\omega)=\sum P(X_1=x_1,X_2=x_2,\ldots,X_n=x_n)\log P(X_1=x_1,X_2=x_2,\ldots,X_n=x_n). \tag{9.64}$$

It is straightforward to prove [68] that a sufficient condition for this limit to exist is i.i.d. random variables, in which case,

$$H=\lim_{n\to\infty}\frac{1}{n}H^{(n)}(\omega)=\lim_{n\to\infty}\frac{1}{n}nH(\omega_1)=H(\omega_1). \tag{9.65}$$

The Shannon–McMillan–Breiman theorem [68] states more generally that for a finite-valued *stationary* stochastic process $\{X_n\}$, this limit exists and converges to the entropy rate H.

If the stochastic system is really a dynamical system as described above, one in which a natural invariant measure μ describes the behavior of typical trajectories, then we attain a direct correspondence of the information theoretic description to the dynamical system in terms of its symbolization. We may develop the so-called **metric entropy**, also known as **Kolmogorov Sinai entropy** (**KS entropy**), h_{KS} [183]. Assuming a more general topological partition,

$$\mathcal{P}=\{A_i\}_{i=0}^k, \tag{9.66}$$

of $k+1$ components, then the resulting entropy of the stochastic process is

$$H(\mathcal{P})=-\sum_{i=0}^{k}\mu(A_i)\ln\mu(A_i). \tag{9.67}$$

However, we wish to build the entropy of the stochastic process of the set theoretic **join**[134] of successive refinements that occur by progressively evolving the dynamical system. Let

$$h(\mu,T,\mathcal{P}) = \lim_{n\to\infty} \frac{1}{n}\left(\bigvee_{i=0}^{n} \mathcal{P}^{(n)}\right), \tag{9.68}$$

where we define

$$\mathcal{P}^{(n)} = \bigvee_{i=0}^{n} T^{-n}(\mathcal{P}), \tag{9.69}$$

and T^{-1} denotes the possibly many-branched preimage if T is not invertible. Thus, the join $\bigvee_{i=0}^{n} T^{-n}(\mathcal{P})$ is the set of all set intersections of the form

$$A_{i_1} \cap T^{-1}(A_{i_2}) \cap \cdots \cap T^{-n}(A_{i_{n+1}}), \quad 0 \le i_k \le n. \tag{9.70}$$

Now we should interpret the meaning of these quantities. H for a stochastic process is the limit of the Shannon entropy of the joint distributions. Literally, it is an average time density of the average information in a stochastic process. A related concept of entropy rate is an *average conditional entropy rate,*

$$H'(X) = \lim_{n\to\infty} H(X_n|X_{n-1},\ldots,X_1). \tag{9.71}$$

Whereas $H(X)$ is an entropy per symbol, $H'(X)$ can be interpreted as the average entropy of seeing the next symbol conditioned on all the previous symbols. There is an important connection which occurs for a stationary process. Under the hypothesis of a stationary stochastic process, there is a theorem [68] that states

$$H(X) = H'(X), \tag{9.72}$$

which further confirms the existence of the limit, Eq. (9.71).

Thus by this connection between entropy rates in dynamical systems we can interpret $h(\mu,T,\mathcal{P}^{(i)})$ in Eq. (9.68) as the information gained per iterate averaged over the limit of long time intervals; call this $h_\mu(T)$ for short. The details of the entropy depend on the **chosen partition** $\mathcal{P}$. As was discussed in Section 6.4.6 and highlighted in Fig. 6.33, the value of entropy measured in a dynamical system depends on the chosen partition. This is most obvious in the extreme case that $\mathcal{P} = P_0$ is defined to be a partition of a single element covering the whole space, in which case all possible symbol sequences of all possible orbits consist of one symbol stream, $0.000\ldots$. This would give zero entropy due to zero surprise, and likewise because $p_0 = 1 \implies \log(1) = 0$. It is natural then to ask if there is a fundamental entropy of the dynamical system, rather than having entropy closely associated with the choice of the partition. From this question follows the quantity

$$h_{KS}(T) \equiv h_\mu(T) = \sup_{\mathcal{P}} h(\mu,T,\mathcal{P}^{(i)}). \tag{9.73}$$

[134]The **join** between two partitions $\mathcal{P}_1 = \{P_1^1, P_1^2, \ldots, P_1^m\}$ and $\mathcal{P}_2 = \{P_2^1, P_2^2, \ldots, P_2^n\}$ is defined $\mathcal{P} = \mathcal{P}_1 \vee \mathcal{P}_2 = (\{P_1^i \cap P_2^j\}_{i=0}^m)_{j=0}^n$, and in general $\mathcal{P} = \bigvee_{i=1}^N \mathcal{P}_i = \mathcal{P}_1 \vee \mathcal{P}_2 \vee \cdots \vee \mathcal{P}_N$ is interpreted successively by repeated application.

This **KS entropy** is the supremum of entropies over all possible partitions. It describes the average bit rate of seeing symbols in terms of all possible partitions and weighted according to natural invariant measure μ. The interpretation of $h_{KS}(T)$ is the description of precision as a degree of surprise of a next prediction with respect to increasing n. When this quantity is positive, in some sense this relates to sensitive dependence on initial conditions.

There is another useful concept of entropy often used in dynamical systems, called **topological entropy** h_{top} [1], which we have already mentioned in Section 6.4.1. We may interpret h_{top} as directly connected to basic information theory and also to h_{KS}. One interpretation of h_{top} is in terms of maximizing entropy by rebalancing the measures so as to make the resulting probabilities extremal. Choosing a simple example of just two states for ease of presentation, we can support this extremal statement by the simple observation that (stating Shannon entropy for two states)

$$(1/2,1/2) = \underset{p}{\text{argmax}}[-p\log p - (1-p)\log(1-p)]. \tag{9.74}$$

This is easily generalized for finer finite partitions. Expanding the same idea to Eq. (9.73) allows us to better understand Parry's maximal entropy measure when it exists. Stated in its simplest terms for a Markov chain, Parry's measure is a rebalancing of probabilities of transition between states so that the resulting entropy of the invariant measure becomes maximal. See [246] and also [53, 140] and [147] for some discussion of how such maximal entropy measures need not generally exist but do exist at least for irreducible subshifts of finite type. Generally the connection between $h_{top}(T)$ and $h_{\mu}(T)$ is formal through the following variational theorem.

Theorem 9.8 (variational principle for entropy—connection between measure theoretic entropy and topological entropy [140, 98, 48]). Given a continuous map $f : M \to M$ on a compact metric space M,

$$h_{top}(f) = \sup_{\mu} h_{\mu}(f), \tag{9.75}$$

where the supremum is taken over those measures μ which are f-invariant Borel probability measures on M.

On the other hand, the direct Definitions 9.13 and 9.14 of topological entropy [246] are in terms of counting numbers of ϵ-separated sets, and how quickly these states of finite precision become separated by iterating the dynamical system; see also [48, 267]. We find the variational principle to be more descriptive than the original definition in terms of understanding the meaning of topological entropy. Further discussion of $h_{top}(T)$ connections with computational methods are made in Section 9.6.

9.6 Computational Estimates of Topological Entropy and Symbolic Dynamics

In Section 9.4, the connection between orbits of a dynamical system and a dynamical system as an entropy process is discussed by example with demonstration of the information present in distinguishing orbits. Further, the connection between measurable dynamics and topological dynamics can be understood in terms of the variational principle for the entropy

theorem, Theorem 9.8. In the discussion of symbolic dynamics in Chapter 6, especially in Section 6.4.1, we discussed symbolic dynamics in depth, including formulas describing entropy in terms of cardinality, Eq. (6.74), and also a related spectral formula, Eq. (6.67). In this section, we will reprise this discussion of topological entropy associated with a dynamical system in more detail and in the context of the formal information theory of this chapter.

9.6.1 Review of Topological Entropy Theory

Adler, Konheim, and McAndrew introduced topological entropy in 1965 [1] in terms of counting the growth rate of covers of open sets under the dynamical system. However, the Bowen definition [44, 46] is in terms of ϵ-separated sets.

Definition 9.13 (**(n,ϵ)-separated [267]**). Given a metric space (M,d) and a dynamical system on this space, $f : M \to M$, a subset $S \subset M$ is (n,ϵ)-separated if

$$d_{n,f}(\mathbf{x},\mathbf{y}) > \epsilon \tag{9.76}$$

for each distinct $\mathbf{x},\mathbf{y} \in S$, $\mathbf{x} \neq \mathbf{y}$, where

$$d_{n,f}(\mathbf{x},\mathbf{y}) = \sup_{0 \leq j < n} d(f^j(\mathbf{x}), f^j(\mathbf{y})). \tag{9.77}$$

In terms of the metric topology[135] this can be roughly described as enumerating the coarsening associated with how open sets are maximally cast across open sets. By "cast across" we simply mean that iterates of the set have images intersecting other open sets. Topological entropy is defined in terms of counting the growth rate of the iteration time n of different orbits, where difference is in terms of an ϵ-scale, as time grows, and then in the limit as this ϵ-coarse grain decreases.

Definition 9.14 (**topological entropy [44, 267]**).

$$h_{top}(f) = \lim_{\epsilon \to 0} \limsup_{n \to \infty} \frac{\log(s(n,\epsilon,f))}{n}, \tag{9.78}$$

where

$$\log(s(n,\epsilon,f)) = \max\{\#(S) : S \subset M \text{ is } (n,\epsilon)\text{- separated by } f\}. \tag{9.79}$$

In this sense, topological entropy is a counting of the rate of new states which develop under the dynamics in terms of states on each coarse scale in the scale limit.

This definition cannot be considered as practical as a computable quantity. Other methods must be deployed. When the dynamical system is a right resolvent sofic shift (see Definition 6.5) we have the simple spectral formula (6.67) to simply and exactly compute topological entropy, which we restate [46, 267]

$$h_{top}(\Sigma'_N) = \ln \rho(A). \tag{9.80}$$

$\rho(A)$ is the spectral radius of the associated transition matrix A, meaning the largest eigenvalue of the matrix of 0's and 1's corresponding to allowed transitions on the generated

[135] A **metric topology** is a topology (see topological space 4.3) whose basis of open sets is simply open balls $N_\epsilon(\mathbf{x})$ defined in terms of a metric d on M. $N_\epsilon(\mathbf{x}) = \{\mathbf{y} : d(\mathbf{x},\mathbf{y}) < \epsilon\}$.

directed graph G_A denoting allowed words in the subshift Σ'_N. Recall that following Definition 6.4, there is a finite-sized transition matrix A that generates a graph G_A that presents the grammar of the subshift Σ'_N of a fullshift Σ_N on N symbols with a corresponding Bernoulli mapping $s : \Sigma'_N \to \Sigma'_N$. This formula is practical in general dynamical systems in the scenario of a Markov partition, Definitions 4.1 and 4.2, from which often a finite graph may be associated with transitions of the corresponding dynamical system meant to properly represent transitions of the associated shift map, and using Theorems 9.10 and 9.11 for the equivalence. We have used this computational method for the logistic map in [29, 41] and the Henon map in [194, 41], to name a few. We wish to highlight two perspectives on this approximation by a Markov model.

- A nested sequence of (imbedded chaotic saddle) subsets allows Markov models of the attractor as a whole, as in Example 9.1.

- In terms of the uniform norm, a given dynamical system, a dynamical system may exist which is exactly Markov, and finite-dimensional computations are exact. Therefore, one way to discuss entropy approximations of a given dynamical system is in terms of sequences of such Markov estimates and understanding the density of the representation.[136]

Another useful description of the topological entropy is the expression, with hypothesis in [47],

$$h_{top}(f) = \limsup_{n\to\infty} \frac{\log(P_n)}{n}, \tag{9.81}$$

where P_n *denotes the number of fixed points of* f^n.[137] Remembering that the popular Devaney definition of chaos (Definition 6.1) includes the requirement that periodic orbits are dense [12, 94], there is some sense in measuring the complexity of the dynamical system in terms of these orbits. A practical perspective is an observation that unstable periodic orbits (**UPO**s) form the "skeleton" of nonlinear dynamics ([7], Cvitanovic, 1988). To this end, [7, 5] have progressed in estimates of thermodynamic properties by using just the short orbits. Numerous theoretical systems were shown to be well described through this approach [5]. Furthermore, recent work using interval arithmetic [130, 129, 128] and computer-assisted proof [132] has allowed for rigorous entropy estimates with this perspective. See also [77] for a discussion of computer-validated proof of symbolic dynamics, and [222, 223, 224], which are based on methods of computational homology [179]. At the forefront and quite impressive, these methods can even be used in infinite-dimensional PDE systems validating the finite-dimensional Galerkin representation as ODEs and the persistence of their periodic orbits despite the finite-dimensional approximation [326, 78, 79].

The use of formula (9.81) is generally to empirically check for convergence for a general mapping, and we emphasize that this formula does not require symbolization as

[136] For example, in the special case of the set of all skew tent maps, a theorem can be found in [19] that states that a given map either is Markov or may be uniformly (sup-norm) estimated to arbitrary accuracy by a map that is Markov; in this sense the family of Markov maps is dense in the uniform topology. See further discussion in Section 4.4 and especially Theorem 4.2.

[137] It is important to count correctly: Given a map $x' = f(x)$, a periodic orbit $\{x_0 x_1, \ldots, x_{n-1}\}$ is n-points which are roots of $f^n(x) = x$. However, not all roots of $f^n(x) = x$ are period n since the period of a point is defined as the *smallest* m such that $f^m(x) = x$. For example, a fixed point ($n = 1$) solves $f(x) = x$, and it also solves $f^2(x) = x$ (and $f^n(x) = x$ for all $n > 0$), but it is called period-1 since 1 is the smallest such n.

shown in Example 9.1 below. However, under the assumption that the mapping is a shift mapping, this formula is similar to a mathematically well-founded statement,

$$h_{top}(\Sigma'_N) = \limsup_{n\to\infty} \frac{\log w_n}{n}, \tag{9.82}$$

where w_n *is the number of words of length* n in Σ'_N. The principled use of both formulas (9.80) and (9.82) are confirmed by the following theorem.

Theorem 9.9 (**see [267]**)**.** A subshift with the Bernoulli-shift map dynamical system, $s : \Sigma'_N \to \Sigma'_N$, has topological entropy that may be computed generally by Eq. (9.80) or Eq. (9.82) when sofic and right resolvent (see comment 6.5).

Remark 9.2. Understanding the difference between the periodic orbits estimate (9.81) and the word count formula (9.82) may be described in terms of symbolizing orbits with a generating partition. When the periodic orbit estimate (9.81) is used specifically for a system that is already symbolized, we interpret this as counting symbolized periodic orbits. Let

$$u_n = \sigma_0.\sigma_1\sigma_2\ldots\sigma_{n-1}, \quad \sigma_i \in \{0,1,\ldots,N-1\}, \tag{9.83}$$

be a word segment of length n of the N symbols associated with Σ_N. Then by definition w_n *is the number of such blocks of* n *bits* that appear in points σ in the subshift, $\sigma \in \Sigma'_N$. These word segments may be part of periodic orbit, in which case that word segment is repeated,

$$\sigma = u_n u_n \ldots \equiv \sigma_0.\sigma_1\sigma_2\ldots\sigma_{n-1}\sigma_0.\sigma_1\sigma_2\ldots\sigma_{n-1}\ldots, \tag{9.84}$$

or it would not be repeated if the point is not part of a periodic orbit. So in the symbolized case, the difference between the two formulae is that generally we expect

$$P_n \leq w_n, \tag{9.85}$$

but the hope is that for large n, the difference becomes small.

Remark 9.3. The key to the practical and general use of the periodic orbit formula (9.81) is that symbolization is not necessary. This is a useful transformation of the problem of estimating entropy, since finding a generating partition is generally a difficult problem in its own right for a general dynamical system [70, 142, 74, 41]. Details of the misrepresentation of using a partition that is not generating are discussed in Section 6.4.6 as a review of [41]. Whether or not we symbolize, the number of periodic orbits has changed, which is why the periodic orbit formula is robust in this sense, although it has the alternative difficulty of being confident that *all* of the periodic orbits up to some large period have been found.

Remark 9.4. The key to algorithmically using the periodic orbit formula (9.81) is the possibility of reliably finding all of the periodic orbits of period n for a rather large n. Given that P_n is expected to grow exponentially, it is a daunting problem to solve

$$g(z) = f^n(z) - z = 0 \tag{9.86}$$

for many roots. By saying many roots, we are not exaggerating, since in our experience [74] to reliably estimate the limit ratio in Eq. (9.81), we have worked with on the order of hundreds of thousands of periodic orbits which are apparently complete lists for the

relatively large $n \simeq 18$. See Example 9.1 using the Ikeda map. When $n = 1$, the obvious approach would be to use a Newton method or variant. And this works even for $n = 2$ or 3 perhaps, but as n increases and P_n grows exponentially, the seeding of initial conditions for the root finder becomes exponentially more difficult as the space between the basins of attraction of each root becomes small. A great deal of progress has been made toward surprisingly robust methods in this computationally challenging problem of pushing the root finding to produce the many roots corresponding to seemingly complete lists of orbits [277, 18, 97, 75].

Example 9.1 (**entropy by periodic orbits of the Ikeda map**)**.** Consider the Ikeda map [173, 155]

$$\begin{aligned} I_{keda} &: R^2 \to R^2 \\ (x,y) \mapsto (x',y') &= (a + b[x\cos(\phi) - y\sin(\phi)], b[x\sin(\phi) + y\cos(\phi)]), \end{aligned} \tag{9.87}$$

where

$$\phi = k - \nu/(1 + x^2 + y^2), \tag{9.88}$$

and we choose parameters $a = 1.0, b = 0.9, k = 0.4$, and $\nu = 6.0$. In [73], the authors conjectured a claim construction of all the periodic orbits through period-22, whereas in [74] we used seemingly all of the roughly 373,000 periodic orbits through period-20 to estimate

$$h_{top}(I_{keda}) \simeq 0.602 < \ln 2 \tag{9.89}$$

by Eq. (9.81). In Fig. 9.7 we show those periodic orbits through period-18. Furthermore in [74] we noted the requirement that a generating partition must uniquely differentiate, by the labeling, all of the iterates on all of the periodic orbits; we used this statement to develop a simple construction to successively symbolize (color) each of the periodic orbits in successively longer periods. As these numbers of periodic orbits swell to tens and hundreds of thousands, the attractor begins to fill out, as seen in Fig. 9.7, to become a useful representation of the symbolic dynamics. Notice an interesting white shadow reminiscent of the stable manifolds that are tangent to the unstable manifolds believed to be associated with generating partitions [70, 142].[138] A thorough study of periodic orbits together with a rigorous computer-assisted proof by interval arithmetic [230] is developed in [129] from which we reprise the table of counted orbits, Table 9.1, including comparable estimates of topological entropy commensurate with our own best $h_{top} \simeq 0.602$.

Example 9.2 (**entropy of the Henon map**)**.** Comparably, for Henon mapping, $h(x,y) = (1 + y - ax^2, bx)$, with $(a,b) = (1.4, 0.3)$ in [129], and finding even more periodic orbits,

$$(n, Q_n, P_n, Q_{\leq n}, P_{\leq n}, h_n) = (30, 37936, 1139275, 109033, 3065317), \tag{9.90}$$

[138] Interestingly, notice that the periodic orbits are expected to distribute roughly according to the invariant measure and thus are more rare at regions of low measure; apparently these "white regions" correspond to just those regions associated by tangencies of stable and unstable manifolds. To see this observation, note that the Sinai–Ruelle–Bowen (SRB) measure is the invariant measure along the closure of all the unstable manifolds of the periodic orbits. A clear shadow of "white" missing periodic orbits (up to the period-18's found) can be seen as transverse curves through the attractor, punctuated at tangency points. This conjecture-observation agrees in principle with the well-accepted conjecture [70, 142] that generating partitions must connect between homoclinic tangencies. See Fig. 6.31 for an explicit construction demonstrating the generating partition for the Henon map constructed directly by this conjecture.

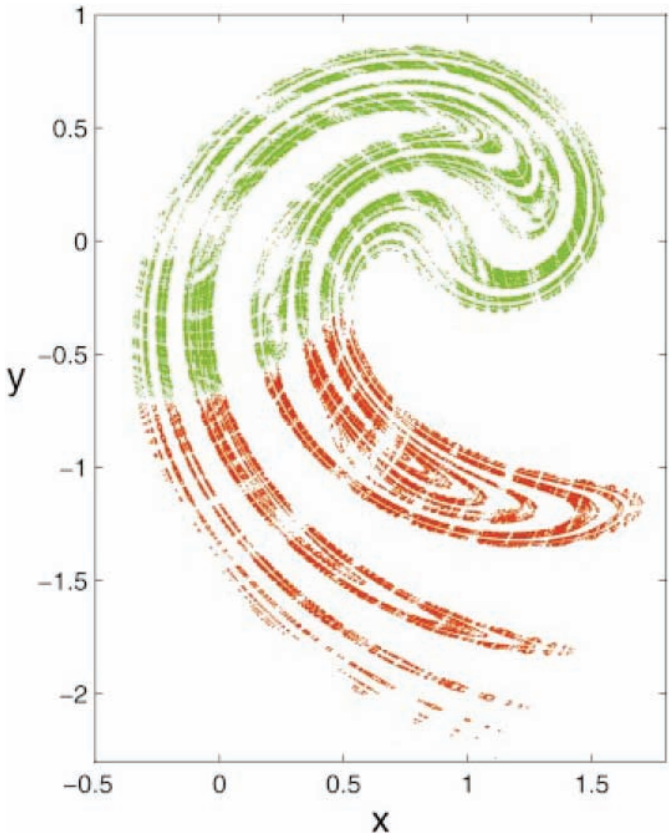

Figure 9.7. *Periodic orbit points up to period-*18 *for the Ikeda–Hammel–Jones–Moloney attractor* (9.87) *from* [74]. *Furthermore, the points on the periodic orbits are colored according to their symbolic representation: green and red dots represent orbit points encoded with symbols* **0** *and* **1**, *respectively. Compare to Table* 9.1 *and Fig*. 9.8. [277]

where rigorous interval arithmetic was used [230]. Compare this to the tabular values for the Ikeda mapping in Table 9.1. Interestingly, but not needed in this symbolic dynamics free computation by periodic orbits; see Fig. 6.31, where a generating partition for this Henon map is constructed directly by locating tangency points.

Example 9.3 (**generating partition and Markov partitions of the Ikeda map**)**.** Consider the Ikeda map, Eq. (9.87). In Fig. 9.7, we show a partition from [74] consistent with a generating partition, this having been constructed by requiring uniqueness of representation for each of the hundreds of thousands of periodic orbits through period-18, and the table of periodic orbits, Table 9.1.

In Fig. 9.8, we show two candidate Markov partitions, each using several symbols from [130]. On the left we see a Markov partition in 4 symbols, and on the right we see a Markov partition in 7 symbols. In [130], a further refined Markov partition in 18 symbols is shown. Generally, a strange attractor may have (infinitely) many embedded Markov partitions representing embedded subshift, where high ordered representations can hope to

Table 9.1. *Periodic orbit counts for the Ikeda–Hammel–Jones–Moloney attractor* (9.87) *from* [74]. Q_n *is the number of periodic orbits of period* n. P_n *is the number of fixed points of the mapping,* f^n. $Q_{\leq n}$ *is the number of cycles of period less than or equal to* n. $P_{\leq n}$ *is the number of fixed points of* f^i *for* $i \leq n$. h_n *is the estimate of the topological entropy for* n, *using Eq.* (9.81). *For comparison, in* [74], *we estimated* $h_{18} \simeq 0.602$. [129]

n	Q_n	P_n	$Q_{\leq n}$	$P_{\leq n}$	h_n
1	2	2	2	2	0.6931
2	1	4	3	4	0.6931
3	2	8	5	10	0.6931
4	3	16	8	22	0.6931
5	4	22	12	42	0.6182
6	7	52	19	84	0.6585
7	10	72	29	154	0.6110
8	14	128	43	266	0.6065
9	26	242	69	500	0.6099
10	46	484	115	960	0.6182
11	76	838	191	1796	0.6119
12	110	1384	301	3116	0.6027
13	194	2524	495	5638	0.6026
14	317	4512	812	10076	0.6010
15	566	8518	1378	18566	0.6033

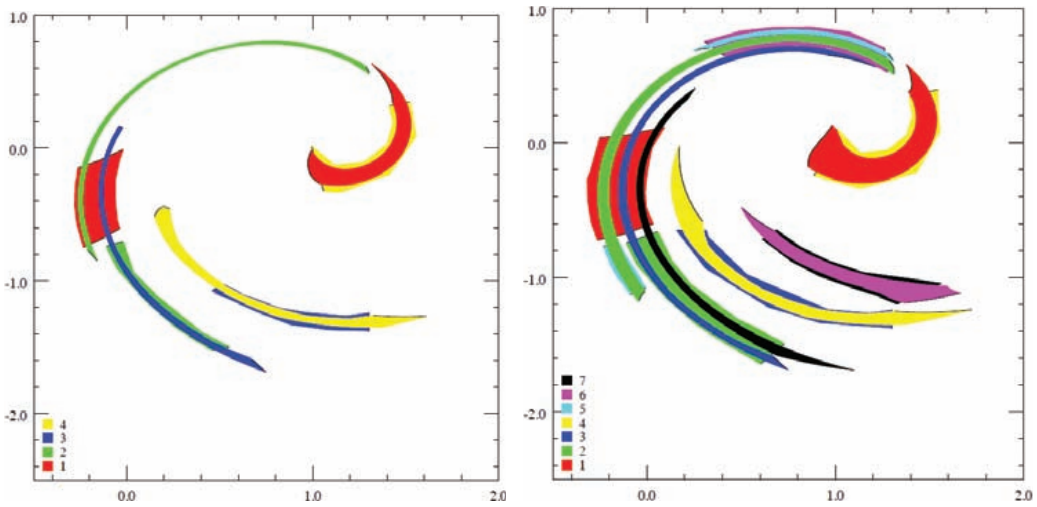

Figure 9.8. *Symbol dynamics of the Ikeda–Hammel–Jones–Moloney attractor* (9.87) *from* [130]. *(Left) Ikeda map,* $\alpha = 6$ *sets* N_i *on which the symbolic dynamics on* 4 *symbols exists and their images. (Right) Ikeda map,* $\alpha = 6$ *sets* N_i *on which the symbolic dynamics on* 7 *symbols exists and their images. Compare to Definition* 4.4 *and Fig.* 4.5.

represent the symbolic dynamics of a greater and greater subset of the full attractor, such as in [19]; compare to Definition 4.4 and Fig. 4.5. Further discussion of the entropy of this attractor is addressed in Example 9.1.

Example 9.4 (entropy by Markov model of the Ikeda map). In Example 9.3, Fig. 9.8, we recall from [130] two Markov model refinements corresponding to two imbedded subshifts Σ_4' and Σ_7' in the Ikeda attractor, using $\alpha = 6$ of Eq. (9.87). See also [129, 128] for techniques of computing enclosures of trajectories, finding and proving the existence of symbolic dynamics, and obtaining rigorous bounds for the topological entropy. From the Ikeda map, the imbedded Markov models yield associated transition matrices:

$$A_4 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \qquad A_7 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{9.91}$$

Using these Markov representations of the transition matrices of the grammars of the imbedded subshifts, together with Eq. (9.80),

$$h_{top}(\Sigma_4') = \ln \rho(A_4) = 0.19946 \quad \text{and} \quad h_{top}(\Sigma_7') = \ln \rho(A_7) = 0.40181, \tag{9.92}$$

and similarly in [130], a further refined 18-symbol Markov model produces

$$h_{top}(\Sigma_{18}') = \ln \rho(A_{18}) = 0.48585. \tag{9.93}$$

We do not show A_{18} here for the sake of space, and in any case, the principle is to continue to refine and therefore to increase the size of the matrices to build ever-increasing refinements. These estimates are each lower bounds of the full entropy, for example,

$$h_{top}(\mathcal{A}) > h_{top}(\Sigma_{18}'), \tag{9.94}$$

where $h_{top}(\mathcal{A})$ denotes the topological entropy on the chaotic Ikeda attractor $\mathcal{A}$ meaning the entropy of the dynamics, (9.87) on this attractor set. We have often used the phrase "embedded subshift" Σ_N' here, by which we mean that there is a subset $\mathcal{A}'$ of the attractor $\mathcal{A}$ such that the subshift Σ_N' is semiconjugate to the dynamics on that subset $\mathcal{A}'$; "imbedded" is the more accurate term.

Theorem 9.10 (comparing topological entropy of factors; see [186]).[139] Suppose there exist two irreducible subshifts of finite type such that Σ_B is a factor of Σ_A; then

$$h_{top}(\Sigma_B) \leq h_{top}(\Sigma_A). \tag{9.95}$$

[139] A subshift of finite type Σ_B is a **factor** (synonym of **semiconjugate**) of another subshift of finite type Σ_A if there exists a continuous and onto mapping $f : \Sigma_A \to \Sigma_B$ that commutes; $s \circ f(\sigma) = f \circ s(\sigma)$. A conjugacy is a special case where f is a homeomorphism.

From this theorem comes the following related statement.

Lemma 9.1 (topological entropy equivalence). Two conjugate dynamical systems have equivalent topological entropy.

Also related is a slightly weaker than conjugacy condition for equivalence of topological entropy.

Theorem 9.11 (topological entropy compared by finite-one semiconjugacy; see [268]). If $g_1 : X \to X$ and $g_2 : Y \to Y$ are two continuous mappings on compact metric spaces X and Y, and $f : X \to Y$ is a semiconjugacy that is uniformly *finite-one*, then

$$h_{top}(g_1) = h_{top}(g_2). \tag{9.96}$$

It is this third theorem that permits us to compute the topological entropy of a dynamical system in terms of its symbolic dynamics.[140] The topological entropy on the attractor is modeled $h_{top}(\mathcal{A}') = h_{top}(\Sigma'_N)$. The phrase "embedded subshift" is rightly often used. The correct word from topology is *imbedding*.[141]

The first theorem explains that in these examples, it is not a surprise that the successive approximations of this example,

$$\Sigma'_4 \hookrightarrow \Sigma'_7 \hookrightarrow \Sigma'_{18}, \tag{9.97}$$

lead to the estimates as found [130, 74],

$$h_{top}(\Sigma'_4) = 0.19946 \leq h_{top}(\Sigma'_7) = 0.40181 \leq h_{top}(\Sigma'_{18}) = 0.48585 \leq h_{top}(I_{keda}) \simeq 0.602. \tag{9.98}$$

The hooked arrow $\hookrightarrow$ denotes imbedding. While not all imbedded Markov models will be comparable between each other, nor will be their entropies. In the case of these examples the nesting explains the reduced entropies. Further discussion can be found of nested imbeddings of horseshoes in homoclinic tangles [227] and chaotic saddles [28].

9.6.2 A Transfer Operator Method of Computing Topological Entropy

Finally, in this section, we briefly mention the possibility of using a transition matrix version of the Ulam–Galerkin matrix to estimate topological entropy, as was studied in detail in [122] and similar to the computation in [29]. In brief, the idea is to use an outer covering of the attractor by "rectangles" or any other regular topological partition such as the successive refinements of Delaunay triangulations [197]. Such an outer covering of the Henon attractor is shown, for example, in Fig. 9.9. As we will discuss, using the transfer operator seems like an excellent and easy method to produce a transfer matrix–based approach toward entropy estimates for a fine grid. After all this follows the same theme as is done in an Ulam method to compute invariant measure or the spectral partitioning methods highlighted earlier in this writing. However, it turns out the approach is not so simple since care

[140] Further note from the theorem that the entropy of a continuous dynamical system, on a compact set, is equal to the entropy of the map restricted to its nonwandering points [268].

[141] A mapping $f : X \to Y$ is defined as an **imbedding** X into Y if a restriction of the range to Z results in a homeomorphism $f : X \to Z$ [233]. Furthermore, when used in the dynamical systems sense and two mappings, when this restriction is with respect to the two mappings $g_1 : X \to X$ and $g_2 : Y \to Y$, and the domain restriction of Z results in a conjugacy, we still use the term imbedding. Often the word embedding may be used in a context that more accurately denotes imbedding as we have defined it here.

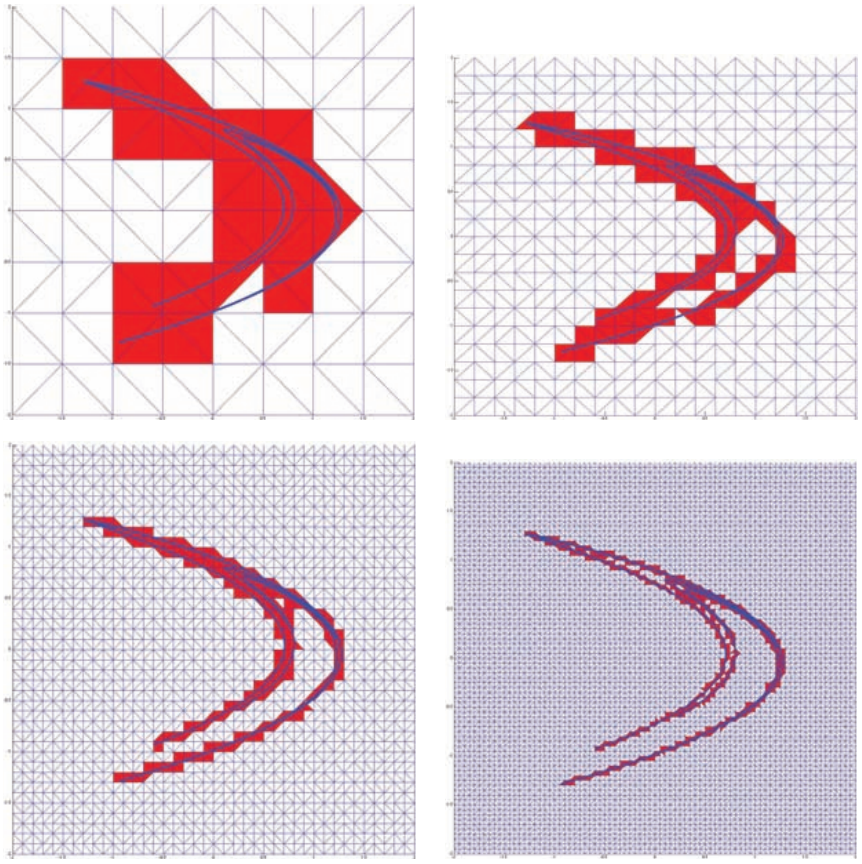

Figure 9.9. *An outer covering (red) of successive partitions by Delaunay triangulations over a Henon attractor. Successively shown are coverings with edge lengths* $h = 0.5, 0.2, 0.1, 0.05$ *resulting in transition matrices* $A_{N\times N}$, $N = 34, 105, 228, 565$, *where* N *is the count of the size of the partition covering the strongly connected component (red) of the full covering.*

must be taken regarding using the generating partition. Since having the generating partition a priori makes any grid method obsolete, we will suggest that transfer matrix–based methods are rarely useful. Nonetheless the theory has some interesting lessons which we will highlight [122, 29].

The idea behind a transfer matrix–based computational method of computing topological entropy hinges on the following refinement theorem relating upper bounds.

Theorem 9.12 (see [122]). Assuming a partition $\mathcal{P}$, let

$$h^*(T,\mathcal{P}) := \lim_{N\to\infty} \frac{\log|w_N(T,\mathcal{P})}{N} = \inf_{N\geq 1} \frac{\log|w_N(T,\mathcal{P})|}{N}; \tag{9.99}$$

then the topological entropy $h_{top}(T)$ of the map T is bounded: If $\mathcal{P}$ is a generating partition (see Definitions 4.5–4.7), then

$$h_{top}(T) \leq \lim_{\operatorname{diam}\mathcal{P}\to 0} \inf h^*(T,\mathcal{P}) \leq h^*(T,\mathcal{P}), \tag{9.100}$$

This theorem provides an upper bound for the topological entropy and suggests a simple constructive algorithm, but one which requires care, as we point out here. We illustrate a direct use of this theorem in Fig. 9.9. In the example in the lower-right frame, $N = 565$ triangular cells cover the attractor. An outer covering of the attractor would result in $h^*(T,\mathcal{P}) \leq \log N$, that we can see is divergent as N is refined. In any case, this is an extreme and not sharp estimate for typical grid refinement.

Example 9.5 (transfer operator for topological entropy for Henon). Direct application of this theorem to the data in Fig. 9.9, building adjacency matrices on the fine Delaunay triangulations,[142]

$$A_{i,j} = \text{if } (1, B_i \to B_j, 0, \text{else}) = ceil(P_{i,j}), \tag{9.101}$$

results in

$$h^* = 1.0123, 1.0271, 1.0641, 1.0245 \tag{9.102}$$

for the specific grid refinements shown,

$$h = 0.5, 0.2, 0.1, 0.05, \tag{9.103}$$

yielding

$$N = 34, 105, 228, 565 \tag{9.104}$$

element coverings, respectively. We see that these are all poor upper bound estimates of a well-regarded $h_{top}(T) \simeq 0.4651$ from [71] derived by methods discussed previously in this section.

So what is wrong? An overestimate is expected, but why does this seemingly obvious and easy approximation method, even with a relatively fine grid, give such large overestimates? The answer lies in the fact that the symbolization is wrong, and not even close. That is, the partition is wrong. $\mathcal{P}$ has 565 elements in our finest cover shown, and we recall that $\log N$ is the upper bound of the entropy of any subshift in Σ_N. While $1.0245 \ll \log 565$, it should not be a surprise that the estimate 1.0245 is not close to 0.4651. Rather the generating partition must be used, and eigenvalues of the transfer matrix are exact if that finite partition happens to be Markov, or close by (9.100) if the partition of the Markov representation is close to generating.

We recall from footnote 138 and Fig. 6.31 that the generating partition for the Henon map connects primary homoclinic tangencies, which is a zig-zag line that turns out to run roughly near $y = 0$. Therefore, to correctly estimate $h_{top}(T)$, it is necessary to associate each cell B_i with a position relative to the generating partition. Clearly, if the generating partition is

$$\mathcal{P}' = (P'_1, P'_2, \ldots, P'_M), \quad M \ll N, \text{ and cell } B_i \subset P'_j, \tag{9.105}$$

then the new symbol j is associated with each such B_i. Similarly for cells B_i which are not entirely within a single partition element, then a decision must be made, perhaps to choose the largest region of overlap, $P'_j \cap B_i$. In this manner, a projection from the larger symbol space is developed,

$$\Pi : \Sigma_N \to \Sigma_M. \tag{9.106}$$

[142] The adjacency matrix A is easily derived from the stochastic matrix corresponding to the Ulam–Galerkin matrix using the $ceil$ function.

The corresponding projected transition matrix should produce the correct topological entropy, but the arbitrarily presented graph cannot be expected to be right-resolvent (see Definition 6.5). The following theorem guarantees that a right-resolvent presentation exists even if the arbitrary projection may not be right-resolvent.

Theorem 9.13 (Lind and Markus [202]). Every sofic shift has a right-resolving presentation.

The proof of this theorem is constructive by the so-called follower method [202], as used in [41] in a context similar to the discussion here, which is to associate new partitions with transition matrices associated with arbitrary partitions. By this method, a new transfer matrix is a right-resolvent presentation of the grammar is developed. Therefore, its corresponding spectral radius is correctly the topological entropy.

However, proceeding in a manner as described above in two steps (i.e., developing a transition matrix associated with a fine partition and then developing a projection to a right-resolvent presentation by the follower construction) may not be considered to be a useful method, since one still must already know the generating partition to properly associate labels to the fine representation of the transfer matrix. As we have already stated, finding the generating partition is a difficult problem. Further, if we already have the generating partition, then simply counting words associated with long orbit segments is a useful and fast-converging method to estimate entropy without needing to resort to the transition matrix, which skips the computational complexity associated with grid-based methods. Furthermore, an exceedingly fine grid would be needed to properly represent the nuance of the w-shaped generating partition seen in Fig. 6.31. For the sake of simplicity, in [122] the authors chose to associate a nongenerating partition as follows: let

$$\mathcal{P}' = (P_1, P2), \text{ where } P_1 = \{(x,y)|y<0\}, \text{ and } P_2 = \{(x,y)|y>0\}. \tag{9.107}$$

Clearly this partition is relatively close to the generating partition. As such, the right-resolvent presentation of the transfer matrix gives an estimate of 0.4628, which we see is less than that from [71],

$$0.4628 < h_{top}(T) \simeq 0.4651. \tag{9.108}$$

That the estimate is close is a reflection of a continuity of the entropy with respect to degree of misplacement discussed in [41, 40]. Furthermore, the theory detailed in [41, 40] shows that using an arbitrary partition risks erratic large errors as emphasized by Figs. 6.32 and 6.33 and Eqs. (6.72)–(6.73) from [41, 40], even if the result is a very interesting devil's staircase–like function describing the consequences of using a nongenerating partition. It could be argued that it is just good luck that $y = 0$, so easily guessed, is in fact close to the generating partition seen in Fig. 6.31 gives a reasonable answer not to be relied upon in general. In any case, the form of the error is not known to be positive or negative, despite the upper bounding statement, Eq. (9.100).

In summary, when estimating topological entropy, the use of transfer operator methods still requires knowledge of the generating partition. Errors will likely be large, as analyzed in [40], if we do not use generating partition information, despite refining grids. However, if we do have the generating partition, then it is perhaps much simpler and more accurate to resort directly to counting words (9.82).

9.7 Lyapunov Exponents, and Metric Entropy and the Ulam Method Connection

In this section we will tighten the connection between metric entropy and how it can be computed in terms of Ulam–Galerkin matrix approximations by considering the Markov action on the corresponding directed graph. This continues in our general theme of connecting concepts from measurable dynamics to computational methods based on transfer operators. Further, we will discuss how this type of computation is exact in the case where a Markov partition is used. Thus, again referring to Section 4.4 and especially Theorem 4.2 concerning density of Markov representations, we can understand a way to analyze the quality of the estimate. Finally, we will review the Pesin identity, which provides a beautiful and deep connection between metric entropy h_{KS} and Lyapunov exponents. We will discuss both estimation and interpretation of these exponents and their information theoretic implications. The main point here is that averaging on single trajectories versus ensemble averages is again the Birkhoff ergodic theorem, Eq. (1.5), which here gives a doubly useful way to compute and understand the same quantities. Compare this section to the introduction with descriptions of two ways of computing Lyapunov exponents discussed in Example 1.4.

9.7.1 Piecewise Linear in an Interval

We start this discussion by specializing to piecewise linear transformations of the interval, specifically to Markov systems that are chaotic; such systems allow the probability density functions to be computed exactly. It is well known that expanded piecewise linear Markov transformations have piecewise constant invariant PDFs, already referred to in Section 4.4.4.

Theorem 9.14 (piecewise constant invariant density; see [50]). Let $\tau : I \to I$ be a piecewise linear Markov transformation of an interval $I = [a,b]$ such that for some $k \geq 1$

$$|(\tau^k)'| > 1,$$

where the derivative exists, which is assumed to be in the interiors of each partition segment. Then τ admits an invariant (probability) density function which is piecewise constant on the partition $\mathcal{P}$ on which τ is Markov.

Using the Frobenius–Perron operator P, the fixed-point function ρ satisfies the definition $P_\tau \rho = \rho$, implying that ρ is the PDF for a measure that is invariant under τ. Since τ is assumed to be a piecewise monotone function, the action of the operator is simply

$$P_\tau \rho(x) = \sum_{z \in \{\tau^{-1}(x)\}} \frac{\rho(z)}{|\tau'(z)|}.$$

The periodic orbit formed by the iteration of $x = a$ forms a partition of the domain $[0,1]$ on which ρ is piecewise constant. On each interval I_i, call the corresponding constant

$$\rho_i = \rho|_{I_i}. \tag{9.109}$$

The probability density function admits an absolutely continuous invariant measure on the Markov partition, the details of which can be found in [50]. For our discussion we

note that this measure can be used to find the Lyapunov exponent, and therefore quantify the average rate of expansion or contraction for an interval under iteration. If we have a Markov partition $\mathcal{P} : 0 = c_0 < c_1 < \cdots < c_{n-1} = 1$, then the Lyapunov exponent Λ is exactly computed as

$$\begin{aligned}\Lambda &= \int_0^1 \ln\left|\tau'(x)\right| \rho(x)\, dx \qquad (9.110)\\ &= \int_{c_0}^{c_1} \ln\left|\tau'(x)\right| \rho_1\, dx + \cdots + \int_{c_{n-2}}^{c_{n-1}} \ln\left|\tau'(x)\right| \rho_{n-1}\, dx\\ &= \ln[\tau'(c_{\frac{1}{2}})] \int_{c_0}^{c_1} \rho_1\, dx + \cdots + [\tau'(c_{n-\frac{1}{2}})] \int_{c_{n-2}}^{c_{n-1}} \ln \rho_{n-1}\, dx\\ &= \sum_{i=1}^{(n-1)} \ln|\tau'(c_{i-\frac{1}{2}})|(c_i - c_{i-1})\rho_i\,.\end{aligned}$$

9.7.2 Nonlinear in an Interval, as a Limit of Piecewise Linear

Given a general transformation of the interval $\tau : I \to I$, which is not assumed to be either Markov or piecewise linear, we may estimate Lyapunov and other measurable and ergodic quantities by refinement in terms of sequences of Markov transformations $\{\tau_n\}$ which uniformly estimate τ. Recall that non-Markov transformations can be written as a weak limit of Markov transformations using Theorem 4.7 of [19], at least in the scenario proved for skew tent maps, as discussed elsewhere in this text.

9.7.3 Pesin's Identity Connects Lyapunov Exponents and Metric Entropy

The famous Pesin entropy identity [250, 321, 191],

$$h_\mu(T) = \sum_{i:\Lambda_i>0} \Lambda_i\,, \qquad (9.111)$$

provides a profound connection between entropy h_{KS} and the (positive) Lyapunov exponents Λ_i, under the hypothesis of ergodicity. In fact, a theorem of Ruelle [272] established

$$h_\mu(T) \le \sum_{i:\Lambda_i>0} \Lambda_i \qquad (9.112)$$

under the hypothesis that T is differentiable and μ is an ergodic invariant measure on a finite-dimensional manifold with compact support. In [108], Eckmann and Ruelle assert that this inequality holds as equality often, but not always for natural measures. However, Pesin proved the equality holds at least if μ is a Lebesgue absolutely continuous invariant measure for a diffeomorphism T [250]. Since then a great deal of work has proceeded in various settings including considerations of natural measure, of the infinite-dimensional setting, and perhaps most interesting in the case of the nonhyperbolic setting of the presence of zero Lyapunov exponents. See [321, 191] for further discussion.

A geometric interpretation of Pesin's entropy formula may be stated as follows. On the one side of the formula, metric entropy describes growth rate of information states with respect to evolution of the dynamics through partitions, as stated directly in Eq. (9.73). However, Lyapunov exponents describe an "average" growth rate of perturbations in characteristic directions of orthogonal successively maximal growth rate directions. Thus we can understand the formula as stating that initial conditions with a given initial precision corresponding to initial hypercubes grow according to the positive exponents in time, thus spreading the initial states across elements of the partition, implying new information generated at a rate descriptive of these exponents. Considering this as an information production process infinitesimally for small initial variations suggests the Pesin formula. Considering further that Lyapunov exponents may be computed in two ways by the Birkhoff formula, either by averaging in time the differential information along "typical" (μ-almost every) initial conditions or by averaging among ensembles of initial conditions but weighting by the ergodic measure μ when it exists, this statement of the Birkhoff ergodic theorem provides two ways of computing and understanding metric entropy. See Eq. (1.5) and Example 1.4. Furthermore, often sampling along a test orbit may provide the simplest means to estimate Lyapunov exponents [318] and hence entropy h_μ according to the Pesin formula. Alternatively, computing Lyapunov exponents by Ulam's method provides another direct method for estimating h_μ through Pesin's identity.

9.8 Information Flow and Transfer Entropy

A natural question in measurable dynamical systems is to ask which parts of a partitioned dynamical system influence other parts of the system. Detecting dependencies between variables is a general statistical question, and in a dynamical systems context this relates to questions of causality. There are many ways one may interpret and computationally address dependency. For example, familiar linear methods such as correlation have some relevance to infer coupling from output signals from parts of a dynamical system, and these methods are very popular especially for their simplicity of application [180]. A popular method is to compute mutual information, $I(X_1; X_2)$ in Eq. (9.35), as a method to consider dynamical influence such as used in [104] in the context of global weather events, as we review in Section 9.9.2. However, both correlation and mutual information more so address overlap of states rather than information flow. Therefore, time dependencies are also missed.

The transfer entropy $T_{J\to I}$ was recently developed by Schreiber [279] to be a statistical measure of information flow, with respect to time, between states of a partitioned phase space in a dynamical system to other states in the dynamical system. Unlike other methods that simply consider common histories, transfer entropy explicitly computes information exchange in a dynamical signal. Here we will review the ideas behind transfer entropy as a measurement of causality in a time evolving system. We present here our work in [31] on this subject. Then we will show how this quantity can be computed using estimates of the Frobenius–Perron transfer operator by carefully masking the resulting matrices.

9.8.1 Definition and Interpretations of Transfer Entropy

To discuss transfer entropy in the setting of dynamical systems, suppose that we have a partitioned dynamical systems on a skew product space $X \times Y$,

$$T : X \times Y \to X \times Y. \tag{9.113}$$

This notation of a single dynamical system with phase space written as a skew product space allows a broad application, as we will highlight in the examples, and helps to clarify the transfer of entropy between the X and Y states. For now, we will further write this system as if it is two coupled dynamical systems having x and y parts describing the action on each component and perhaps with coupling between components.

$$T(x,y) = (T_x(x,y), T_y(x,y)), \tag{9.114}$$

where

$$\begin{aligned} T_x : X \times Y &\to X \\ x_n &\mapsto x_{n+1} = T_x(x_n, y_n), \end{aligned} \tag{9.115}$$

and likewise

$$\begin{aligned} T_y : X \times Y &\to Y \\ y_n &\mapsto y_{n+1} = T_y(x_n, y_n). \end{aligned} \tag{9.116}$$

This notation allows that $x \in X$ and $y \in Y$ may each be vector (multivariate) quantities and even of different dimensions from each other. See the caricature of this arrangement in Fig. 9.10.

Let

$$x_n^{(k)} = (x_n, x_{n-1}, x_{n-2}, \ldots, x_{n-k+1}) \tag{9.117}$$

be the measurements of a dynamical system T_x, at times

$$t^{(k)} = (t_n, t_{n-1}, t_{n-2}, \ldots, t_{n-k+1}), \tag{9.118}$$

sequentially. In this notation, the space X is partitioned into states $\{x\}$, and hence x_n denotes the measured state at time t_n. Note that we have chosen here not to index in any way the partition $\{x\}$, which may be some numerical grid as shown in Fig. 9.10, since subindices are already being used to denote time, and superindices denote time-depth of the sequence discussed. So an index to denote space would be a bit of notation overload. We may denote simply x, x', and x'' to distinguish states where needed. Likewise, $y_n^{(k)}$ denotes sequential measurements of y at times $t^{(k)}$, and Y may be partitioned into states $\{y\}$ as seen in Fig. 9.10.

The main idea leading to transfer entropy will be to measure the **deviation from the Markov property,** which would presume

$$p(x_{n+1}|x_n^{(k)}) = p(x_{n+1}|x_n^{(k)}, y_n^{(l)}), \tag{9.119}$$

that the state $(x_{n+1}|x_n^{(k)})$ does not include dependency on $y_n^{(l)}$. When there is a departure from this Markovian assumption, the suggestion is that there is no information flow as conditional dependency in time from y to x. The measurement of this deviation between these two distributions will be by a conditional Kullback–Leibler divergence, which we will build toward in the following.

The joint entropy[143] of a sequence of measurements written in the notation of Eqs. (9.117)–(9.118) is

$$H(x_n^{(k)}) = -\sum_{x_n^{(k)}} p(x_n^{(k)}) \log p(x_n^{(k)}). \tag{9.120}$$

[143] Definition 9.6.

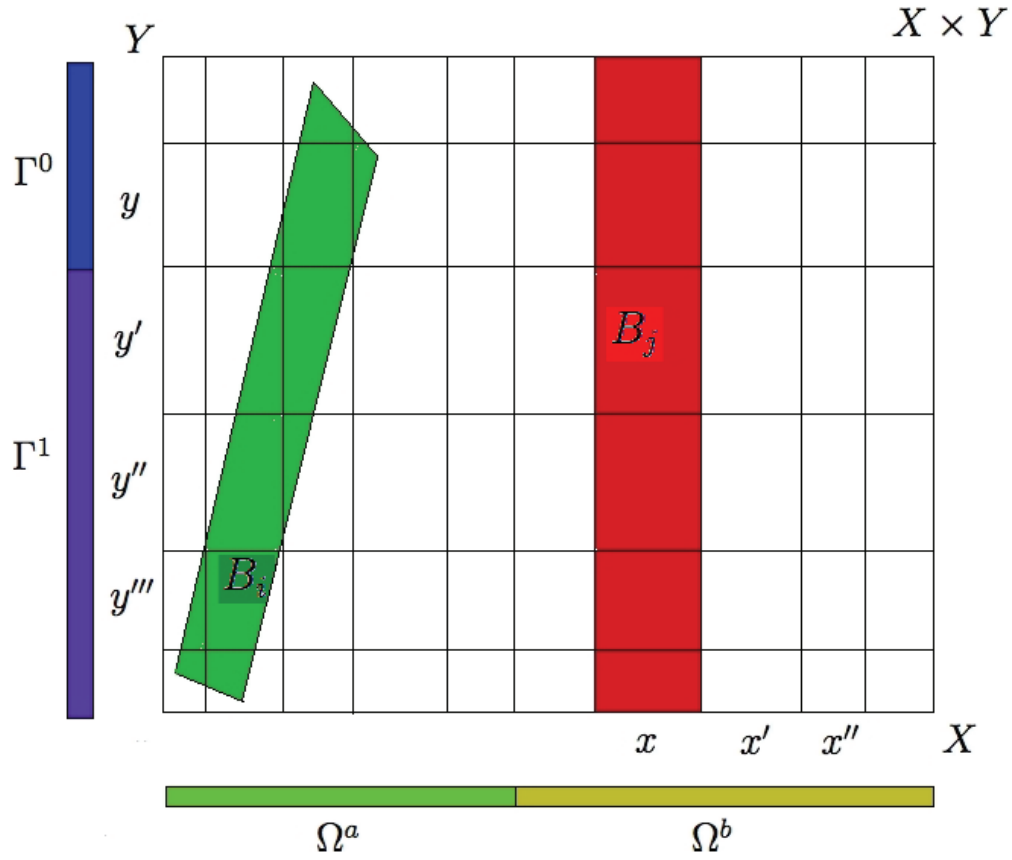

Figure 9.10. *In a skew product space $X \times Y$, to discuss transfer entropy between states $\{x\}$ a partition of X and states $\{y\}$ of Y, some of which are illustrated as x, x', x'' and y, y', y'', y'''. A coarser partition $\{\Omega^a, \Omega^b\}$ of X in symbols a and b and likewise $\{\Gamma^0, \Gamma^1\}$ of Y in symbols* 0 *and* 1 *are also illustrated*. [31]

A conditional entropy,[144]

$$\begin{aligned} H(x_{n+1}|x_n^{(k)}) &= -\sum p(x_{n+1}, x_n^{(k)}) \log p(x_{n+1}|x_n^{(k)}) \\ &= H(x_{n+1}, x_n^{(k)}) - H(x_n^{(k)}) \\ &= H(x_{n+1}^{(k+1)}) - H(x_n^{(k)}), \end{aligned} \tag{9.121}$$

is approximately an *entropy rate*,[145] which as it is written quantifies the amount of new information that a new measurement of x_{n+1} allows following the k-prior measurements, $x_n^{(k)}$. Note that the second equality follows the probability chain rule,

$$p(x_{n+1}|x_n^{(k)}) = \frac{p(x_{n+1}^{(k+1)})}{p(x_n^{(k)})}, \tag{9.122}$$

[144] Definition 9.7.

[145] This is an entropy rate in the limit $k \to \infty$ according to Definition 9.11.

and the last equality from the notational convention for writing the states,

$$(x_{n+1},x_n^{(k)}) = (x_{n+1},x_n,x_{n-1},\ldots,x_{n-k+1}) = (x_{n+1}^{(k+1)}). \tag{9.123}$$

Transfer entropy is defined in terms of a Kullback–Leibler divergence, $D_{KL}(p_1||p_2)$, from Definition 9.9 but adapted for the conditional probabilities:[146]

$$D_{KL}(p_1(A|B)||p_2(A|B)) = \sum_{a,b} p_1(a,b)\log\frac{p_1(a|b)}{p_2(a|b)}. \tag{9.124}$$

The states are specifically designed to highlight transfer of entropy between the states X to Y (or vice versa Y to X) of a dynamical system written as a skew product, Eq. (9.113) Define [279],

$$T_{x\to y} = \sum p(x_{n+1},x_n^{(k)},y_n^{(l)})\log\frac{p(x_{n+1}|x_n^{(k)},y_n^{(l)})}{p(x_{n+1}|x_n^{(k)})}, \tag{9.125}$$

which we see may be equivalently written as a difference of entropy rates, like conditional entropies:[147]

$$T_{y\to x} = H(x_{n+1}|x_n^{(l)}) - H(x_{n+1}|x_n^{(l)},y_n^{(k)}). \tag{9.126}$$

The key to computation is joint probabilities and conditional probabilities as they appear in Eqs. (9.126) and (9.129). There are two major ways we may make estimates of these probabilities, but both involve coarse-graining the states. A direct application of formulae (9.120) and (9.121), and likewise for the joint conditional entropy, to Eq. (9.125) allows

$$T_{y\to x} = [H(x_{n+1},x_n) - H(x_n)] - [H(x_{n+1},x_n,y_n) - H(x_n,y_n)], \tag{9.127}$$

which serves as a useful method of direct computation.

This may be a most useful form for computation, but for interpretation, a useful form is in terms of a conditional Kullback–Leibler divergence,

$$T_{y\to x} = D_{KL}(p(x_{n+1}|x_n^{(k)},y_n^{(l)})||p(x_{n+1}|x_n^{(k)})), \tag{9.128}$$

found by putting together Eqs. (9.124) and (9.125). In this form, as already noted in Eq. (9.119), transfer entropy has the interpretation as a measurement of the deviation from the Markov property, which would be the truth of Eq. (9.119). That the state $(x_{n+1}|x_n^{(k)})$ does not include dependency on $y_n^{(l)}$ suggests that there is no information flow as a conditional dependency in time from y to x causing an influence on transition probabilities of x. In this sense, the conditional Kullback–Leibler divergence (9.128) describes the deviation of the information content from the Markovian assumption. In this sense, $T_{y\to x}$ describes an information flow from the marginal subsystem y to marginal subsystem x. Likewise, and asymmetrically,

$$T_{x\to y} = H(y_{n+1}|y_n^{(l)}) - H(y_{n+1}|x_n^{(l)},y_n^{(k)}), \tag{9.129}$$

[146] Recall that the Kullback–Leibler of a single random variable A with probability distribution is an error-like quantity describing the entropy difference between the true entropy using the correct coding model $\log p_1(A)$ versus a coding model $\log p_2(A)$ with a model distribution $p_2(A)$ of A. Thus, conditional Kullback–Leibler is a direct application for conditional probability $p_1(A|B)$ with a model $p_2(A|B)$.

[147] Again, these become entropy rates as $k,l\to\infty$, as already discussed in Eq. (9.121).

and it is immediate to note that generally

$$T_{x\to y} \neq T_{y\to x}. \tag{9.130}$$

This is not a surprise both on the grounds that it has already been stated that Kullback–Leibler divergence is not symmetric; also, there is no prior expectation that influences should be directionally equal.

A partition $\{z\}$ serves as a symbolization which in projection by Π_x and Π_y is the grid $\{x\}$ and $\{y\}$, respectively. It may be more useful to consider information transfer in terms of a coarser statement of states. For example, see Fig. 9.10, where we represent a partition Ω and Γ of X and Y, respectively. For convenience of presentation we represent two states in each partition:

$$\Omega = \{\Omega^a, \Omega^b\} \quad \text{and} \quad \Gamma = \{\Gamma^0, \Gamma^1\}. \tag{9.131}$$

In this case, the estimates of all of the several probabilities can be summed in a manner just discussed above. Then the transfer entropy $T_{x\to y}$ becomes in terms of the states of the coarse partitions. The question of how a coarse partition may represent the transfer entropy of a system relative to what would be computed with a finer partition has been discussed in [149], with the surprising result that the direction of information flow can be effectively measured as not just a poor estimate by the coarse partition, but possibly even of the wrong sign. [31]

9.9 Examples of Transfer Entropy and Mutual Information in Dynamical Systems

9.9.1 An Example of Transfer Entropy: Information Flow in Synchrony

In our recent paper [31], we chose the perspective that synchronization of oscillators is a process of transferring information between them. The phenomenon of synchronization has been found in various aspects of nature and science [298]. It was initially perhaps a surprise when it was discovered that two, or many, oscillators can oscillate chaotically, but if coupled appropriately they may synchronize and then oscillate identically even if all chaotic together; this is a description of the simplest form of synchronization which is of identical oscillators. See Fig. 9.11. Applications have ranged widely from biology [300, 139] to mathematical epidemiology [159], and chaotic oscillators [248], communication devices in engineering [69], etc. Generally, the analysis of chaotic synchronization has followed a discussion of stability of the synchronization manifold, which is taken to be the identity function when identical oscillators [247], or some perturbation thereof for nonidentical oscillators [302], often by some form of master stability function analysis.

Considering as we have reviewed in this text that chaotic oscillators have a corresponding symbolic dynamics description, then coupling must correspond to some form of exchange of this information. Here we describe our perspective in [31] of coupled oscillators as sharing information, and then the process of synchronization is one where the shared information is an entrainment of the entropy production. In this perspective, when oscillators synchronize, it can be understood that they must be sharing symbols in order that they may each express the same symbolic dynamics. Furthermore, depending on the degree of cocoupling, or master-slave coupling or somewhere in between, the directionality

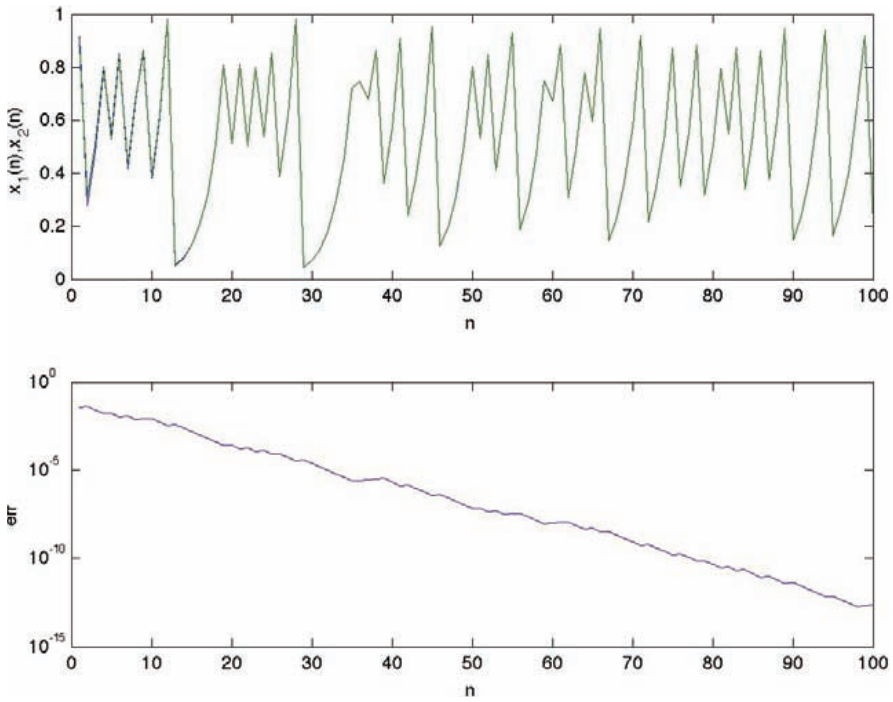

Figure 9.11. *In a nonlinearly coupled skew tent map system, Eq.* (9.133)*, of identical oscillators,* $a_1 = a_2 = 0.63$*, and master-slave configuration,* $\delta = 0.6$*,* $\epsilon = 0.0$ *(parameters as in* [157]*). Note (above) how the signals entrain and (below) the error,* $error(n) = |x(n) - y(n)|$*, decreases exponentially.* [157]

of the information flow can be described by the transfer entropy. A study of anticipating synchronization with a transfer entropy perspective in the note of studying the appropriate scale necessary to infer directionality is found in [149].

Consider the following skew tent map system as an example coupling element to highlight our discussion [157], which is a full folding form [19], meaning two-one:

$$f_a(x) = \left\{ \begin{array}{ccc} \frac{x}{a} & \text{if} & 0 \leq x \leq a \\ \frac{1-x}{1-a} & \text{if} & a \leq x \leq 1 \end{array} \right\}. \tag{9.132}$$

Let us couple these in the following nonlinear manner [157]:

$$\left(\begin{array}{c} x_{n+1} \\ y_{n+1} \end{array} \right) = \mathbf{G} \left(\begin{array}{c} x_n \\ y_n \end{array} \right) = \left(\begin{array}{c} f_{a_1}(x_n) + \delta(y_n - x_n) \\ f_{a_2}(y_n) + \epsilon(x_n - y_n) \end{array} \right). \tag{9.133}$$

Note that written in this form, if $a_1 = a_2$ and $\epsilon = 0$ but $\delta > 0$, we have a master-slave system of identical systems, as illustrated in Fig. 9.11, where we see a stable synchronized identity manifold with error decreasing exponentially to zero. On the other hand, if $\epsilon = \delta$ but $a_1 \neq a_2$, we can study symmetrically coupled but nonidentical systems in Fig. 9.12. There, the identity manifold is not exponentially stable but is apparently a Lyapunov-stable manifold, since the error, $error(n) = |x(n) - y(n)|$, remains small for both scenarios shown in the figures, $a_1 = 0.63$ but $a_2 = 0.65$ and $a_2 = 0.7$, respectively, with progressively larger

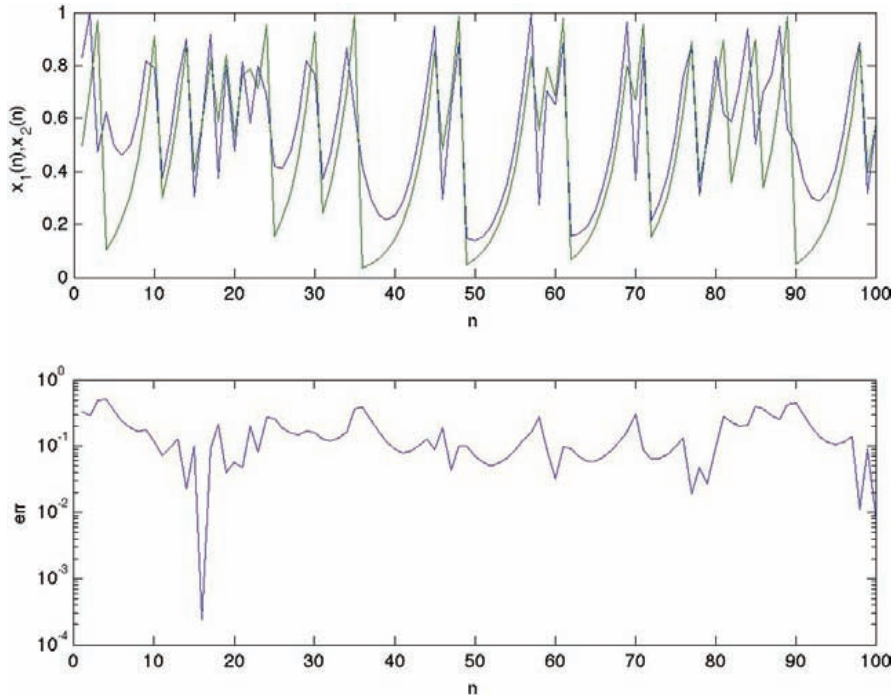

Figure 9.12. *A nonlinearly coupled skew tent map system, Eq.* (9.133), *of nonidentical oscillators,* $a_1 = 0.63, a_2 = 0.65$, *and master-slave configuration,* $\delta = 0.6$, $\epsilon = 0.0$. *Note (above) how the signals approximately entrain and (below) the error,* $error(n) = |x(n) - y(n)|$, *decreases close to zero, where it remains close to an identity manifold,* $x = y$, *where it is stable in a Lyapunov stability sense.* [31]

but stable errors. Our presentation here is designed to introduce the perspective of transfer entropy to understand the process of synchronization in terms of information flow, and from this perspective to gain not only an idea of when oscillators synchronize but perhaps if one or the other is acting as a master or a slave. Furthermore, the perspective is distinct from a master stability formalism.

With coupling resulting in various identical and nonidentical synchronization scenarios as illustrated in Fig. 9.12, we will analyze the information transfer across a study of both parameter matches and mismatches and across various coupling strengths and directionalities. In Figs. 9.13 and 9.14, we see the results of transfer entropies, $T_{x\to y}$ and $T_{y\to x}$, respectively, in the scenario of identical oscillators $a_1 = a_2 = 0.63$ for coupling parameters being swept $0 \leq \delta \leq 0.8$ and $0 \leq \epsilon \leq 0.8$. We see that due to the symmetry of the form of the coupled systems, Eq. (9.133), the mode of synchronization is opposite as expected. When $T_{x\to y}$ is relatively larger than $T_{y\to x}$, then the interpretation is that relatively more information is flowing from the x system to the y system, and vice versa. This source of communication is due to coupling the formulation of synchronization. Large changes in this quantity signal the sharing of information leading to synchronization.

In the asymmetric case, $0.55 \leq a_1, a_2 \leq 0.65$, we show a master-slave coupling $\epsilon = 0$, $\delta = 0.6$ in Fig. 9.15 and compare to Figs. 9.11 and 9.12. In the master-slave scenario chosen, the x-oscillator is driving the y-oscillator. As such, the x-oscillator is sending its

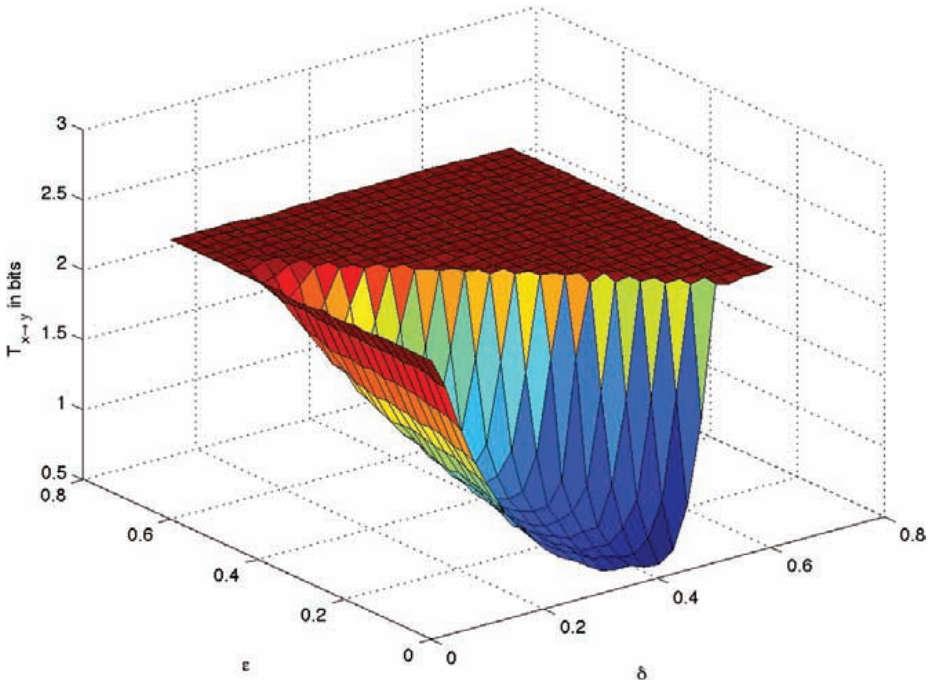

Figure 9.13. *Transfer entropy, $T_{x\to y}$ measured in bits, of the system* (9.133)*, in the identical parameter scenario, $a_1 = a_2 = 0.63$, which often results in synchronization depending on the coupling parameters swept, $0 \leq \delta \leq 0.8$ and $0 \leq \epsilon \leq 0.8$ as shown. Contrast to $T_{y\to x}$ shown in Fig.* 9.14*, where the transfer entropy clearly has an opposite phase relative to the coupling parameters,* (ϵ,δ). [31]

states in the form of bits to the y-oscillator as should be measured that $T_{x\to y} > T_{y\to x}$ when synchronizing and more so when a great deal of information "effort" is required to maintain synchronization. This we interpret as what is seen in Fig. 9.15 in that when the oscillators are identical, $a_1 = a_2$ shown on the diagonal, the transfer entropy difference $T_{x\to y} > T_{y\to x}$ is smallest since the synchronization requires the smallest exchange of information once started. In contrast, $T_{x\to y} > T_{y\to x}$ is largest when the oscillators are most dissimilar, and we see in Fig. 9.13 how "strained" the synchronization can be, since the error cannot go to zero as the oscillators are only loosely bound.

9.9.2 An Example of Mutual Information: Information Sharing in a Spatiotemporal Dynamical System

Whereas transfer entropy is designed to determine direction of information flow, mutual information $I(X_1; X_2)$ in Eq. (9.35) is well suited to decide the simpler question as to whether there is simply a coupling in a large and complex dynamical system. The advantage of using the simpler but perhaps less informative measure, as it does not give directionality, is that it may require less data.

A recent and exciting application of mutual information comes from an important spatiotemporal dynamical system from analysis of global climate [104], as seen in Fig. 9.16.

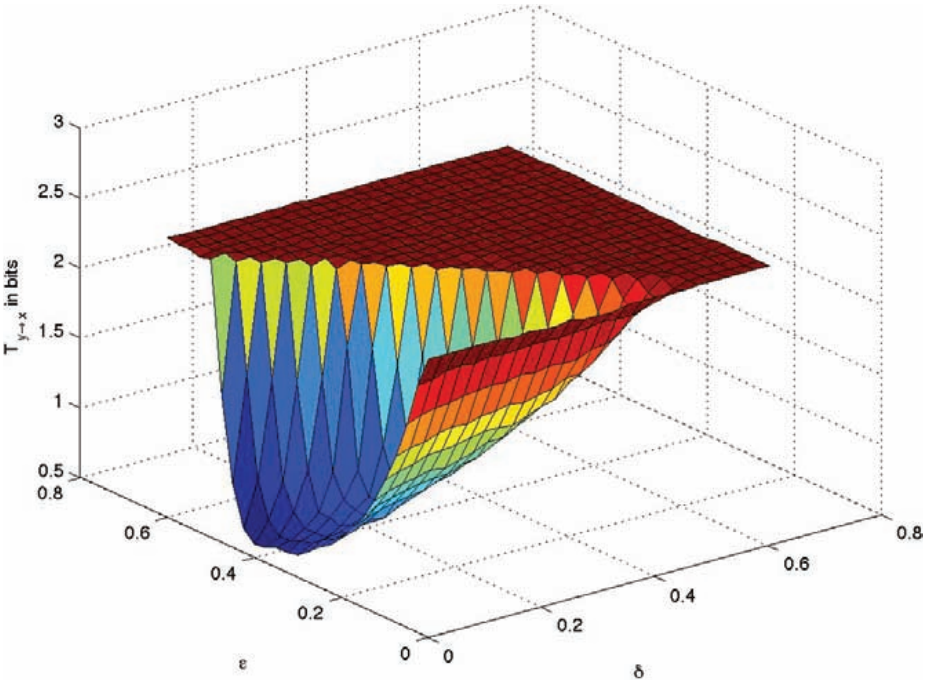

Figure 9.14. *Transfer entropy, $T_{y\to x}$ measured in bits, of the system* (9.133) *in the identical parameter scenario, $a_1 = a_2 = 0.63$, which often results in synchronization depending on the coupling parameters swept, $0 \le \delta \le 0.8$ and $0 \le \epsilon \le 0.8$ as shown. Compare to $T_{x\to y}$ shown in Fig.* 9.13. [31]

The study in [104] used a monthly averaged global SAT field to capture the complex dynamics in the interface between ocean and atmosphere due to heat exchange and other local processes. This allowed the study of atmospheric and oceanic dynamics using the same climate network. They used data provided by the National Center for Environmental Prediction/National Center for Atmospheric Research (NCEP/NCAR) and model output from the World Climate Research Programmes (WCRPs) Coupled Model Intercomparison Project phase 3 (CMIP3) multimodel data set.

These spatiotemporal data can be understood as a time series, $x_i(t)$, at each spatial site i modeled on the globe. Pairs of sites, i, j, can be compared for the mutual information in the measured values for states in the data $x_i(t)$ and $x_j(t)$ leading to $I(X_i; X_j)$. Following a thresholding decision leads to a matrix of couplings $A_{i,j}$ descriptive of mutual information between sites on the globe. The interpretation is that the climate at sites with large values recorded in $A_{i,j}$ are somehow dynamically linked. In Fig. 9.16, we illustrate what was shown in [104], which is a representation of the prominence of each site i on the globe colored according to that prominence. The measure of prominence shown is the vertex betweenness centrality, labeled BC_i. Betweenness centrality is defined as the total number of shortest paths[148] in the corresponding undirected graph which pass through the

[148] In a graph, $G = (V, E)$ consisting of the set of vertices and edges which are simply vertex pairs, a path between i and j of steps along edges of the graph which connect a pair of vertices. A shortest path between i and j is a path which is no longer than any other path between i and j.

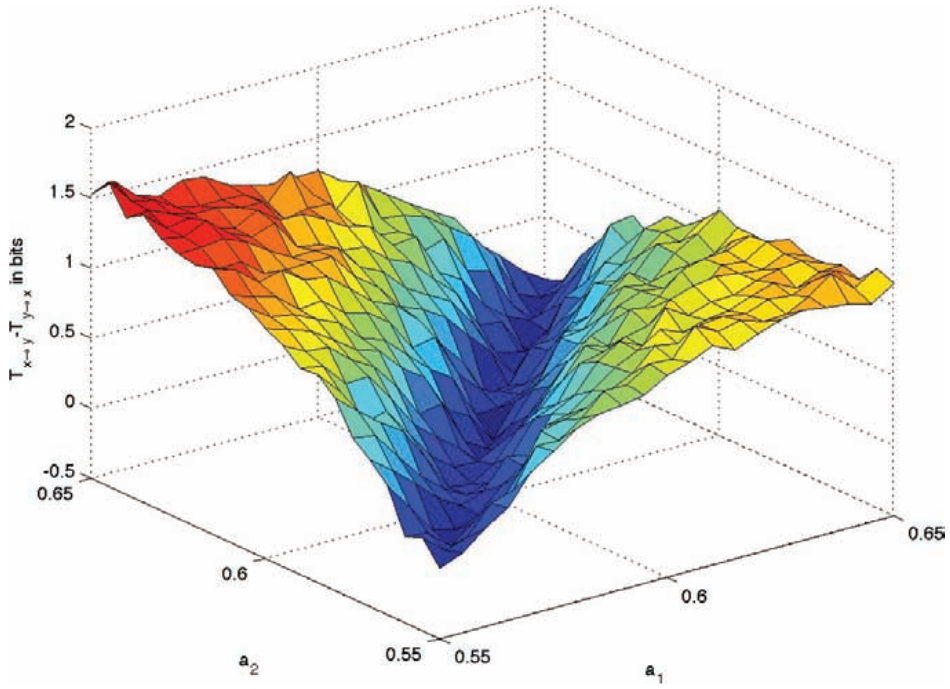

Figure 9.15. *Transfer entropy difference, $T_{y\to x} - T_{y\to x}$, measured in bits, of the system* (9.133) *in the nonidentical parameter scenario sweep,* $0.55 \leq a_1, a_2 \leq 0.65$, *and master-slave coupling* $\epsilon = 0$, $\delta = 0.6$. *Compare to $T_{x\to y}$ shown in Fig.* 9.13. *Contrast to $T_{y\to x}$ shown in Fig.* 9.14, *where the transfer entropy clearly has an opposite phase relative to the coupling parameters,* (ϵ, δ). *Also compare to Figs.* 9.11 *and* 9.12. [31]

vertex labeled i. BC_i can be considered as descriptive as to how important the vertex i is to any process running on the graph. Since the graph is built by considering mutual information, it may be inferred that a site i with high BC_i is descriptive of a dynamically important site in the spatiotemporal process, in this case the process being global climate. It is striking how this information theoretic quantification of global climate agrees with known oceanographic processes as shown in Fig. 9.16.

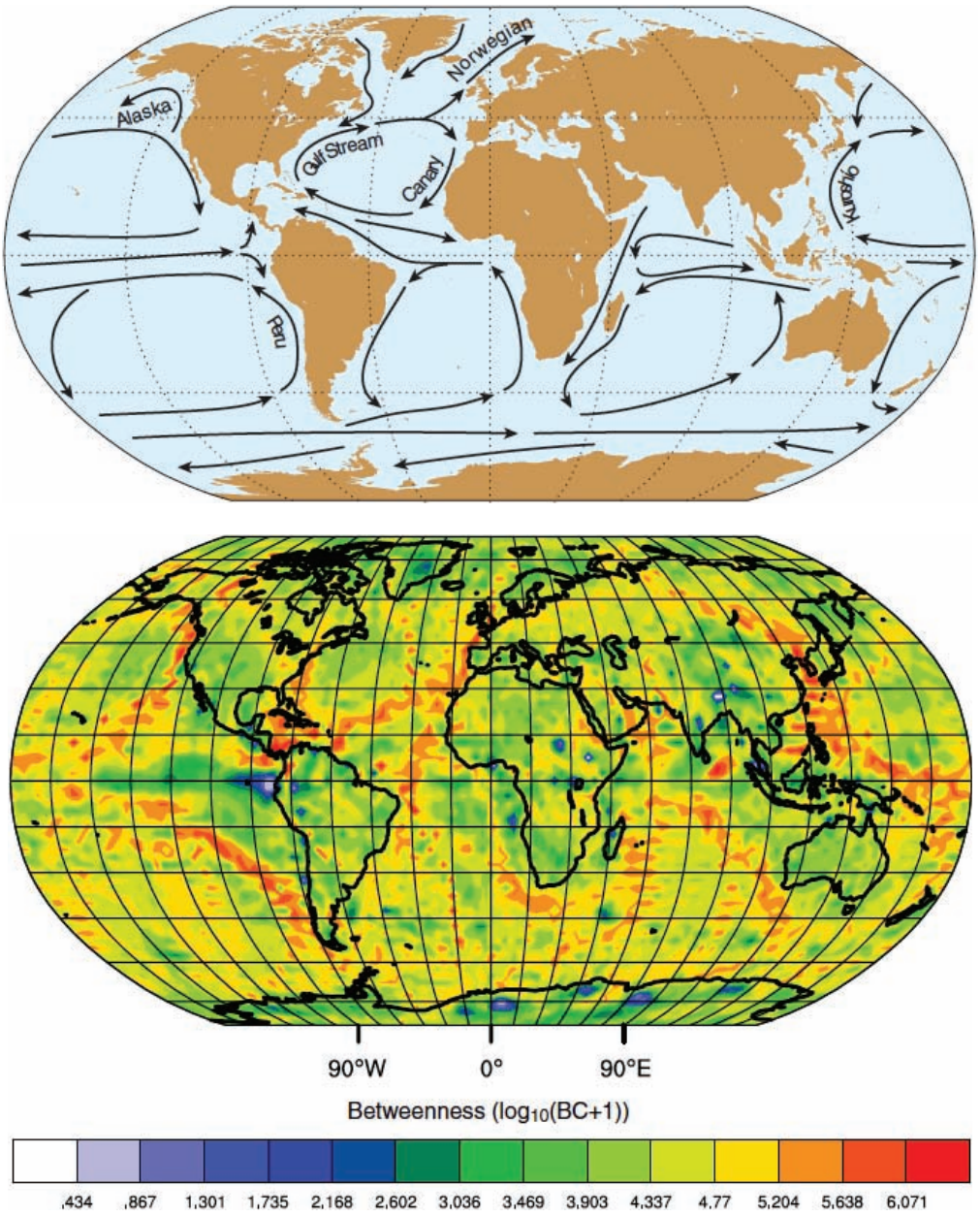

Figure 9.16. *Mutual information mapping of a global climate network from* [104]. *(Top) Underlying and known global oceanographic circulations. (Bottom) Betweenness centrality BC_i from a network derived from mutual information $I(X_i;X_j)$ between the global climate data from spatial sites i, j across the globe. Note that the mutual information theoretic quantification of the global climate is agreeable with the known underlying oceanographic processes.*