A Brief Introduction to:

Information Theory, Excess Entropy

and

Computational Mechanics

April 1998

(Revised October 2002)

David Feldman
College of the Atlantic
105 Eden Street
Bar Harbor, ME 04609
dave@hornacek.coa.edu
http://hornacek.coa.edu/dave/

Contents

Chapter 1

Background in Information Theory

In this chapter I'll introduce some of the essential ideas and quantities from information theory. The material reviewed here is standard. A good, thorough reference is the text by Cover and Thomas [8]. I find this text to be a excellent blend of rigor and qualitative reasoning. The original paper [43] by the founder of information theory, Claude Shannon has been reprinted in [44]. Ref. [44] also contains a very nice, mostly qualitative introduction to information theory by Shannon and Weaver. Shannon's papers have been collected in Ref. [46]. The statistical mechanics textbook by Robertson [40] contains a nice discussion of Shannon's information in the context of statistical mechanics. In general I like Robertson's approach, but sometimes in his book it's hard to see the forest for the trees. Baierlein's text [2] also discusses statistical mechanics from an information theory point of view. His discussion of probability and entropy is excellent and he does a nice job motivating the definition of the Shannon entropy. The range of statistical mechanics topics that he covers is not very modern, however. Another introduction to information theory is that of Pierce [38]. This has a very high word to equation ratio. I've only glanced at it, but it seems quite good.

1.1 Notation

In the following, I shall use capital letters to indicate a discrete random variable, and lowercase letters to indicate a particular value of that variable. For example, let X be a random variable. The variable X may take on the values $x \in \mathcal{X}$. Here $\mathcal{X}$ is the finite set of all possible values for X and is

referred to as the *alphabet* of X.

The probability that X takes on the particular value x is written $\Pr(X = x)$, or just $\Pr(x)$. We may also form joint and conditional probabilities. Let Y be another random variable with $Y = y \in \mathcal{Y}$. The probability that $X = x$ and $Y = y$ is written $\Pr(X = x, Y = y)$, or $\Pr(x, y)$ and is referred to as a joint probability. The conditional probability that $X = x$ given $Y = y$ is written $\Pr(X = x|Y = y)$ or simply $\Pr(x|y)$.

1.2 Shannon Entropy and its Many Interpretations

1.2.1 Entropy as Uncertainty

The use of probabilities to describe a situation implies some uncertainty. If I toss a fair coin, I don't know what the outcome will be. I can, however, describe the situation with a probability distribution: $\{\Pr(\text{Coin} = \text{Heads}) = 1/2, \Pr(\text{Coin} = \text{Tails}) = 1/2\}$. If the coin is biased, there is a different distribution: $\{\Pr(\text{BiasedCoin} = \text{Heads}) = 0.9, \Pr(\text{BiasedCoin} = \text{Tails}) = 0.1\}$.

All probability distributions are not created equal. Some distributions indicate more uncertainty than others; it is clear that we are more in doubt about the outcome of the fair coin than the biased coin. The question before us now is: can we make this notion of uncertainty or doubt quantitative? That is, can we come up with some mathematical entity that takes a probability distribution and returns a number that can be interpreted as a measure of the uncertainty associated with that distribution.

Let's proceed by considering what features such a measure should have. For concreteness, let's call this measure $H[X]$. That is, H takes the probability distribution of X $X = \{\Pr(1), \Pr(2), , \cdots \Pr(N)\}$ and returns a real number. The picture here is that there are N possible values X can assume, and $\Pr(i)$ is the probability that X equals the i^{th} possible value.

First, we surely want H to be maximized by a uniform distribution. After all, a uniform distribution corresponds to complete uncertainty. Everything is equally likely to occur — you can't get much more uncertain than that.

Second, it seems reasonable to ask that H is a continuous function of the probabilities. An arbitrarily small change in the probabilities should lead to an arbitrarily small change in H.

Third, we know that we can group probabilities in different ways. For

example, consider a variable X with the following distribution

$$X = \{\, \Pr(X = A) = .5, \Pr(X = B) = .2, \Pr(X = C) = .3 \,\} \,. \tag{1.1}$$

One way to view this distribution is that outcome C *or* B occurs half of the time. When it does occur, outcome B occurs with probability .4. That is:

$$\begin{aligned} X &= \{\, \Pr(X = A) = .5, \Pr(X = Y) = .5, \,\} \,, & (1.2) \\ Y &= \{ \Pr(Y = B) = .4, \Pr(Y = C) = .6 \,\} \,. & (1.3) \end{aligned}$$

We would like the uncertainty measure H not to depend on what sort of grouping games we play. In other words, we want H to be a function of the distribution itself and not a function of how we group events within that distribution.

Remarkably, the above three requirements are enough to determine the form of H *uniquely* up to a multiplicative constant.

1.2.2 Axiomatic Definition

Let's state the above three requirements more carefully and generally. Let $H(p)$ be a real-valued function of $\Pr(1), \Pr(2), \cdots, \Pr(N)$. Let the following three requirements hold:

1. $H(\, \Pr(1), \Pr(2), \cdots, \Pr(N)\,)$ reaches a maximum when the distribution is uniform; $\Pr(i) = 1/N \;\forall\; i$.

2. $H(\, \Pr(1), \Pr(2), \cdots, \Pr(N)\,)$ is a continuous function of the $\Pr(i)$'s.

3. The last requirement is awkward to write mathematically, but no less intuitive than the first two. As mentioned above, the idea is that we want H to be independent of how we group the probabilities of individual events into subsets. I'll follow the notation of Robertson [40]. Let the N probabilities be grouped into k subsets, w_k:

$$w_1 = \sum_{i=1}^{n_1} p_i \;;\; w_2 = \sum_{i=n_1+1}^{n_2} p_i \;;\; \ldots \tag{1.4}$$

   Then, we assume

$$H[p] = H[w] + \sum_{j=1}^{k} w_j H[\{p_i/w_j\}_j] \,, \tag{1.5}$$

   where the notation $\{p_i/w_j\}_j$ indicates that the sum extends over those p_i's which make up a particular w_j.

Given the above three requirements, it follows that,

$$H[X] = k \sum_{x \in \mathcal{X}} \Pr(x) \log \Pr(x) \ , \tag{1.6}$$

where k is an arbitrary constant [8, 40, 44]. The choice of constant amounts to nothing more than a choice of units. For the remainder of this paper, I shall use base 2 logarithms and fix k at -1. The units of $H[X]$ for this choice of constant are called *bits*.

Thus, we define the Shannon entropy of a random variable X by:

$$H[X] \equiv - \sum_{x \in \mathcal{X}} \Pr(x) \log_2(\Pr(x)) \ . \tag{1.7}$$

The notation $H[X]$ can be misleading. $H[X]$ is *not* a function of X! It is a function of the *probability distribution* of the random variable X. The value of $H[X]$ does not depend on whatever value X assumes.

Note that the entropy is never negative. One can easily prove that

$$H[X] \geq 0 \ . \tag{1.8}$$

Also note that $H[X] = 0$ if and only if X is known with certainty: *i.e.*, the probability of one outcome is 1 and the probability of all other outcomes is 0. (To show this one needs to use $\lim_{x \to \infty} x \log_2 x = 0$.)

The axiomatic definition of H given above justifies the following statement: $H(p)$ is *the* quantitative measure of the amount of uncertainty associated with a probability distribution p. But the story does not end here. There are many other ways we can view the Shannon entropy. In the following several sections, we explore some of these additional interpretations.

1.2.3 Shannon Entropy as Thermodynamic Entropy

It is not hard to show that $H(p)$ is equivalent to the usual thermodynamic entropy,

$$S(E) = \log N(E) \tag{1.9}$$

where $N(E)$ is the number of accessible microstates as function of energy E. Since microstates of equal energy are assumed to be equally likely, the probability of the $i^{\underline{th}}$ state occurring is just

$$\Pr(i) \ = \ \frac{1}{N(E)}, \quad \forall \, i \ . \tag{1.10}$$

Plugging Eq. (1.10) into Eq. (1.7), we see immediately that the thermodynamic entropy, Eq. (1.9) results.

It is this connection with thermodynamics that led Shannon to call his uncertainty measure entropy. (Legend has it that he was encouraged to do so by John von Neumann, who said that since no one really understands what entropy is, calling his new measure entropy would give Shannon "a big edge in the debates.")

1.2.4 Shannon Entropy as Average Surprise

Here is another way to view Eq. (1.7): The quantity $-\log_2 \Pr(i)$ is sometimes referred to as the *surprise* associated with the outcome i. If $\Pr(i)$ is small, we would be quite surprised if the outcome actually was i. Accordingly, $-\log_2 \Pr(i)$ is large for small $\Pr(i)$. And if $\Pr(i)$ is large, we see that the surprise is small. So it seems entirely reasonable to call $-\log_2 \Pr(i)$ the surprise.

Thus, we may view Eq. (1.7) as telling us that $H[X]$ is the expectation value of the surprise;

$$H[X] = \sum_x \{-\log_2 \Pr(x)\} \Pr(x) \;=\; \langle\, -\log_2 \Pr(x) \,\rangle \;. \tag{1.11}$$

The entropy tells us, on average, how surprised we will be if we learn the value of the variable X. This observation strengthens the assertion that $H(p)$ is a measure of the uncertainty associated with the probability distribution p. The more uncertain we are about an outcome, the more surprised we will be (on average) when we learn of the actual outcome.

We can also use this line of reasoning to see why H is referred to as information. Let us return to the example of a coin toss. Suppose I told you the outcome of the toss of a fair coin. This piece of information would be quite interesting to you, as before I told you the outcome you were completely in the dark. On the other hand, if it is the biased coin with a 90% probability of heads that is thrown, telling you the outcome of the toss is not as useful. "Big deal" you might say. "I was already pretty sure it was heads anyway; you really haven't given me much information." It is in this sense that $H[X]$ provides a measure of information. The greater $H[X]$, the more informative, on average, a measurement of X is.

1.2.5 Entropy and Yes-No Questions

Entropy is also related to how difficult it is to guess the value of a random variable. This is discussed rather thoroughly and clearly in chapter 5 of

Ref. [8]. Here, I'll just explain the general ideas qualitatively.

We begin with an example. Consider the random variable X with following distribution:

$$\begin{aligned} \{\Pr(X = A) &= 1/2, \Pr(X = B) &= 1/4, \\ \Pr(X = C) &= 1/8, \Pr(X = D) &= 1/8\} \, . \end{aligned} \tag{1.12}$$

On average, how many yes-no questions will it take you to figure out the value of X? Well, your first guess would be $X = A$. You would be right half of the time. Thus, half of the time you'll only need one question to guess correctly. If you guessed incorrectly, your next move would be to guess $X = B$. Again, you will be right half of the time. So, half of the time you'll need to make the $X = B$ guess, and half of the time that guess will be correct. As a result, $1/4$ of the time it will take two guesses to determine X.

If your $X = B$ guess was incorrect, you'll need to make one more guess, say, $X = C$. Regardless of the outcome of this guess, you'll end up knowing the value of X, since if $X \neq C$, it must be that $X = D$. So, half of the time you'll need to make the $X = B$ guess, and half of the time that guess will be wrong, necessitating the $X = C$ guess. Hence, $1/4$ of the time you'll need to make 3 guesses. Adding this up, we have:

$$\text{Average \# of Guesses} = \frac{1}{2}(1) + \frac{1}{4}(2) + \frac{1}{4}(3) = 1.75 \, . \tag{1.13}$$

It turns out that the entropy of the distribution given in Eq. (1.12) is exactly equal to 1.75!

This is not a coincidence. One can show that [8]

$$H[X] \leq \text{Average \# of Yes/No Questions to Determine X} \leq H[X] + 1 \, . \tag{1.14}$$

This result assumes that the guesser is making optimal guesses. That is, roughly speaking at every guess, he or she tries to "divide the probability in half." This is exactly the strategy we employed in the above example.

Eq. (1.14) might appear a little mysterious as first. As a slightly less mysterious example, consider another distribution:

$$\begin{aligned} \{\Pr(Y = \alpha) = &\ 1/4, \Pr(Y = \beta) &= 1/4, \\ \Pr(Y = \gamma) = &\ 1/4, \Pr(Y = \delta) &= 1/4\} \, . \end{aligned} \tag{1.15}$$

Clearly it will take an average of 2 guesses to determine the value of Y. The variable X is easier to guess because a lot of the probability is concentrated on $X = A$ and $X = B$, and we can exploit this in our guessing.

This idea of entropy as the average number of yes-no guesses is consonant with our earlier interpretation of entropy as a measure of uncertainty. The more uncertain we are about an event, the harder it is to guess the outcome.

1.2.6 Entropy and Coding

Let's pause now and consider coding. What is a code? Well, at the simplest level, it's just one thing that stands for another thing. We can code base-10 digits using a hexadecimal alphabet and we can code English letters using Morse code. Sometime we encode an object to make it secret. For example, when I send my credit card number over the internet to order some Peet's coffee beans I make sure that my web browser encrypts the credit card information so that my card number remains a secret to any third parties who may be "listening in". We may also encode an object to "make it smaller" so it can be stored or transmitted efficiently. It is this second type of coding that we will consider here.

How does one devise an efficient code? The idea is to choose short code words for objects that occur most frequently. As an example, consider again the example of Eq. (1.12). It makes sense to use the shortest possible code for the event $X = A$ since it has the highest probability of occurring.

This business of choosing more probable events should be familiar—it's almost identical to the strategy we employed when we were trying to guess what X was. In fact, the process of yes-no guessing specifies a binary code [8]. Stating the mathematical definition of a code is a little cumbersome and subtle and goes beyond the scope of this introduction. (For example, one must deal with the issue of how one knows when one code word ends and another begins.)

Rather than go through the details, let's just consider again the example of Eq. (1.12). Recall our procedure for guessing the outcome of X and consider the sequence of questions that led up to our determining a particular value. To make our code, for each "yes" answer we'll use a 1 and for each "no" answer we'll use a 0. The result is the following code:

$$\begin{aligned} A &\longrightarrow 1 \\ B &\longrightarrow 01 \\ C &\longrightarrow 001 \\ D &\longrightarrow 000 \end{aligned} \tag{1.16}$$

For example, if we discovered that $X = B$ we would have gotten a "no" to our first questions and a "yes" to our second, corresponding to 01.

Given this correspondence between yes-no questions and binary coding, we see that Eq. (1.14) implies that:

$$H[X] \leq \text{Average Length of Binary Code for X} \leq H[X] + 1 \,. \tag{1.17}$$

Before wrapping up this section, I'll state a slightly more technical result. Suppose one is encoding N identically distributed random variables X with a binary code. Then, in the $N \to \infty$ limit:

$$\frac{1}{N}(\text{Average Length of Optimal Binary Code for X}) = H[X] \,. \tag{1.18}$$

This is the famous Shannon source coding theorem.

Each digit in a binary code corresponds to one *bit*, a flip-flop memory device that can be in one of two positions. Thus, Eq. (1.18) tells us that $H[X]$ is the average number of bits needed to store the value of the random variable X.

1.3 Joint and Conditional Entropy

I'll continue by defining some variants of the entropy discussed above. Most of these quantities are quite well named. I'll also state some relationships and properties of these quantities.

First, the *joint entropy* of two random variables, X and Y, is defined in the natural way:

$$H[X,Y] \equiv -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} \Pr(x,y)\log_2(\Pr(x,y)) \,. \tag{1.19}$$

The joint entropy is a measure of the uncertainty associated with a joint distribution.

Next, we define the *conditional entropy*:

$$H[X|Y] \equiv -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} \Pr(x,y)\log_2 \Pr(x|y) \,. \tag{1.20}$$

As one would guess from its name, the conditional entropy measures the uncertainty associated with a conditional probability. Note that $H[X|Y]$ is the expectation value of the conditional surprise, $-\log_2 \Pr(x|y)$ where the average is weighted by the *joint* distribution.

By writing $\Pr(x,y) = \Pr(x)\Pr(y|x)$ and taking the expectation value of the log of both sides of this equation, we see that the joint entropy obeys the following, pleasing chain rule:

$$H[X,Y] = H[X] + H[Y|X] \,. \tag{1.21}$$

There are two noteworthy consequences of this observation. First, we may write

$$H[Y|X] = H[X,Y] - H[X] \; . \tag{1.22}$$

As $H[X] \geq 0$, we obtain the sensible result that conditioning reduces entropy. That is, knowledge of one variable can never increase our uncertainty about other variables. Second, Eq.(1.21) makes it quite clear that

$$H[y|x] \neq H[x|y] \; . \tag{1.23}$$

1.4 Mutual Information

We now turn our attention to mutual information. We define the *mutual information*, $I[X;Y]$, of two random variables X and Y via:

$$I[X;Y] \equiv \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \Pr(x,y) \log_2 \left[\frac{\Pr(x,y)}{\Pr(x)\Pr(y)} \right] \; . \tag{1.24}$$

Some straightforward manipulations show us that

$$\begin{aligned} I[X;Y] &= H[X] - H[X|Y] & (1.25)\\ &= H[Y] - H[Y|X] & (1.26)\\ &= H[Y] + H[X] - H[X,Y] \; . & (1.27) \end{aligned}$$

The above shows quite clearly that $I(X;Y) = I(Y;X)$.

Eq. (1.25) show us why I is called the mutual information; we see that mutual information between two variables is the reduction in uncertainty of one variable due to knowledge of another. If knowledge of Y reduces our uncertainty of X, then we say that Y carries information about X.

Looking at Eq. (1.24), it's not hard to see that $I[X;Y]$ vanishes if X and Y are independently distributed; $\Pr(x,y) = \Pr(x)\Pr(y)$. Also, we see that the mutual information between two variables vanishes if both variables have zero entropy.

1.5 Entropy of Continuous Variables

1.5.1 Continuous Entropy $\longleftrightarrow$ Discrete Entropy

Can Shannon's entropy, Eq. (1.7) be generalized to apply to a continuous variable? The "principle of least astonishment" suggests that

$$H_c[X] \; = \; -\int dx \, \Pr(x) \log_2 \Pr(x) \; . \tag{1.28}$$

It turns out that this is the case. However, we should invoke more rigor than the principle appealed to above.

While the above equation seems logical, it's not as simply obtained from the discrete formula, Eq. (1.7), as one might think. As an example, let's imagine the transition from a discrete partitioning of the unit interval to the interval itself. We can easily form a picture of how to do this by letting the "width" of the discrete partitions get smaller and smaller. However, as we do this, the number of partitions grows. Thus, the number of variables (partitions) diverges and the entropy diverges as well.

So, defining the entropy for the case of a continuous variable requires some care. In the following careful discussion, I follow closely the exposition in Cover and Thomas [8]. Let us consider a random variable X with a probability distribution $f(x)$. We can then divide the range of X into discrete bins of width Δ. We discretize x by forming the variables X^Δ, defined by:

$$X^\Delta \, = \, x_i \; , \quad \text{if } i\Delta \leq X < (i+1)\Delta \; . \tag{1.29}$$

The notation here is that X^Δ refers to the discrete variables that can take on the values x_i. The numbers x_i are a particular value, to be chosen below, in the interval. The probability that $X^\Delta = x_i$ is obtained simply by integrating the probability density over the appropriate interval:

$$\Pr(X^\Delta = x_i) \, \equiv p_i \, = \, \int_{i\Delta}^{(i+1)\Delta} f(x) dx \; . \tag{1.30}$$

We now harken back to the days of Calc I. According to the mean value theorem, within each of these bins of length Δ there exists some x_i such that

$$f(x_i)\Delta \, = \, \int_{i\Delta}^{(i+1)\Delta} f(x) dx \; . \tag{1.31}$$

Let's choose to use the above x_i's as our x_i's in Eq. (1.29). Thus, we can combine the above two equations and write

$$p_i \, = \, f(x_i)\Delta \; . \tag{1.32}$$

We are now may use these probabilities to write down the discrete entropy, Eq. (1.7), of our discretized variable X^Δ:

$$H[X^\Delta] \;\; = \;\; -\sum_i p_i \log_2 p_i \tag{1.33}$$

$$-\sum_i f(x_i)\Delta \log_2 [f(x_i)\Delta] \tag{1.34}$$

$$-\sum_i f(x_i)\Delta \log_2 f(x_i) - \sum_i f(x_i)\Delta \log_2 \Delta \qquad (1.35)$$

$$-\sum_i f(x_i)\Delta \log_2 f(x_i) - \log_2 \Delta \; . \qquad (1.36)$$

The last equality follows from the normalization of the distribution $f(x)$. Again hearkening back to Calc I, we notice that as the width of our bins vanishes, the first term approaches the integral of $f(x)\log_2 f(x)$. Thus,

$$H[X^\Delta] \longrightarrow H[X] - \log_2 \Delta, \quad \text{as } \Delta \longrightarrow 0 \; . \qquad (1.37)$$

Where H_c is the entropy for a continuous random variable as defined in Eq. (1.28). Note that $-\log_2 \Delta$ diverges as Δ vanishes. This is exactly the explosion anticipated above associated with the infinite uncertainty of an infinite number of variables.

The moral of the above story is that the entropy of a continuous random variable does not equal the entropy of the discretized random variable in the limit that the bin size goes to zero. The inequality arises because when taking the limit, the number of variables goes to infinity and thus has infinite entropy. If this divergence associated with vanishing bin sizes is subtracted away, then one obtains the entropy for a continuous variable.

1.5.2 Careful Definition

Let's restate the definition of the entropy of a continuous variable slightly more carefully and then examine a few of its properties. Following Cover and Thomas [8, p. 224], let X be a random variable with a cumulative distribution $F(x) \equiv \Pr(X \leq x)$. The variable X is said to be continuous if the function $F(x)$ is continuous. The probability density function for X is given by $f(x) \equiv F'(x)$, provided that $f(x)$ is normalized. Those values of x for which $f(x) \neq 0$ are referred to as the *support set* of X.

We then define the differential entropy of a continuous random variable X as:

$$H_c[X] \equiv -\int f(x)\log_2 f(x)\, dx \; . \qquad (1.38)$$

The integration is understood to be over the support set of X.

The differential entropy behaves somewhat differently than its discrete cousin. Most notably, H_c can be negative. For example, consider a variable uniformly distributed on the interval $(0, b)$. The probability density function is $1/b$ and the entropy is $\log_2 b$. Clearly if $b < 0$ we'll have negative entropy.

The differential entropy is unchanged by a shift of variable. That is,

$$H_c[Y + l] = H_c[Y] \; . \qquad (1.39)$$

However, rescaling the variable does change the entropy;

$$H_c[bX] \;=\; H_c[X] + \log_2 |b| \;. \tag{1.40}$$

More generally, if $\vec{X}$ is a vector-valued variable and A is some matrix, one can show that [8, p. 232]

$$H_c[A\vec{X}] \;=\; H_c[\vec{X}] + \log_2 |\det A| \;. \tag{1.41}$$

Differential conditional entropy and mutual information are defined in the obvious ways. While the differential entropy can be negative, the differential mutual is still non-negative:

$$I_c[X;Y] \geq 0 \;. \tag{1.42}$$

It is also comforting to note that the differential information between the continuous variables X and Y *is* equal to the limit of the discretized versions of X and Y in the limit that the bin sizes go to zero. Thus, there is no need for the subscript c indicating that the variables are continuous and it will subsequently be omitted.

If two continuous variables are simultaneously rescaled by the same factor, their mutual information is unchanged;

$$I[aX;aY] \;=\; I[X;Y] \;. \tag{1.43}$$

Indeed, one would be distressed if the mutual information did not have this property.

I conclude this section by mentioning, as Cover and Thomas do, that all of these above formulae hold only if the integrals exist. This leads to some interesting existential thoughts. (Does $x = y$ if neither x nor y exist?) Putting these thoughts on hold for a later time, we now proceed to the next chapter to define and discuss the excess entropy.

Chapter 2

Entropy Density and Excess Entropy

In this chapter we apply the ideas of Chapter 1 to infinite strings of symbols. Entropy density is a standard quantity; a discussion of it can be found in most texts on information theory and many texts on dynamical systems. Excess entropy is not a standard quantity. To my knowledge, it has not been discussed in any texts. A brief history of excess entropy isfound below in sec. 2.3.

There are several relatively recent review articles on excess entropy and entropy convergence: e.g., Refs. [18, 20, 49]. Those wishing to go into greater depth are urged to consult those reviews. The goal of this chapter, then, is to present the main ideas behind excess entropy and entropy convergence so that the reader finds these somewhat more technical review articles accessible.

2.1 Entropy Density

Let's begin by fixing some notation. Consider an infinite string:

$$\overleftrightarrow{S} = \ldots S_{-1} S_0 S_1 S_2 \ldots \tag{2.1}$$

chosen from some finite alphabet, $S_i = s_i \in \mathcal{A}$. We may view this sequence of variables as being a time series of measurements, the symbolic dynamics from some map, or the configurations of a statistical mechanical spin chain. We denote a block of L consecutive variables as $S^L = S_1, \ldots, S_L$. Let $\Pr(s_i, s_{i+1}, \ldots, s_{i+L}) = \Pr(s^L)$ denote the joint probability over blocks of

L consecutive symbols. We shall assume translational invariance:

$$\Pr(\, s_i, s_{i+1}, \cdots, s_{i+L}\,) \;=\; \Pr(\, s_1, s_2, \cdots, s_L\,) \;\; \forall\, i,\, L\;. \tag{2.2}$$

Equivalently, this requirement means that the symbols may be viewed as having been generated by a stationary stochastic process.

We may divide this infinite string into a left half ("past") $\overleftarrow{S}$, and a right half ("future") $\overrightarrow{S}$, as follows:

$$\overleftarrow{S} \;\equiv \cdots S_{-3}\, S_{-2}\, S_{-1}\;, \tag{2.3}$$

and

$$\overrightarrow{S} \;\equiv S_0\, S_1\, S_2 \cdots\;. \tag{2.4}$$

We would like to measure the entropy of the string $\overleftrightarrow{S}$. How can we go about doing this? Well, we can start by figuring out the entropy of blocks of adjacent variables within $\overleftrightarrow{S}$. Let the Shannon entropy of a block of L consecutive symbols be denoted by $H(L)$:

$$H(L) \equiv -\sum_{s_1 \in \mathcal{A}} \cdots \sum_{s_L \in \mathcal{A}} \Pr\left(s_1, \ldots, s_L\right) \log_2 \Pr\left(s_1, \ldots, s_L\right)\;. \tag{2.5}$$

To determine the entropy of the entire system $\overleftrightarrow{S}$, we could take the $L \to \infty$ limit. It is not hard to see, however, that $H(L)$ will diverge as L goes to infinity. After all, as L goes to infinity we're trying to keep track of an infinite number of variables, so it certainly seems reasonable that $H(L)$ will also be infinite.

This divergence is a drag; we would like to be able to compare the entropy of different infinite strings of random variable, yet this will be hard if the entropy is infinite. Fortunately, there's a natural solution to this problem; we form an entropy density:

$$h_\mu \equiv \lim_{L \to \infty} \frac{H(L)}{L}\;. \tag{2.6}$$

The quantity h_μ goes by different names depending on the area of application. If we view $\overleftrightarrow{S}$ as a spatially-extended system such as a one-dimensional Ising system, h_μ is known as the entropy density or the entropy per site. If we view $\overleftrightarrow{S}$ as a time series or as a discrete signal being transmitted across, say, a telegraph line, h_μ would be called the *entropy rate*. In dynamical systems parlance, h_μ is known as the *metric entropy*.

It is perhaps not immediately obvious that the limit in Eq. (2.6) exists. I won't prove the existence of the limit here, but I will try to make its existence moderately plausible. As the length of our block of variables grows, the probability of any one particular L-block tends to decrease exponentially. For example, if the variables are independently chosen by a fair coin, then $\Pr(s^L) = 2^{-L} \quad \forall \quad s^L$. As a result, $\log_2[\Pr(s^L)] \sim L$. Plugging this into Eq. (2.6), we see that the limit will exist. More rigorously, one can show that h_μ exists (at a minimum) for all stationary stochastic processes [8].

The entropy density can also be written in terms of a conditional entropy:

$$h_\mu = \lim_{L\to\infty} H[S_L|S_0S_1\ldots S_{L-1}]\,. \tag{2.7}$$

Thus, h_μ is the uncertainty of the distribution over L-blocks of spins conditioned on the first $(L-1)$ spins in that block. These two expressions are shown in Sec. (A.1) to be equivalent.

Eq. (2.7) provides us with another interpretation of h_μ; it is the entropy, or uncertainty, associated with a given symbol if all the preceding symbols are known. Put another way, the entropy density provides an answer to the question: given the knowledge of *all* the previous symbols, how uncertain are you, on average, about the next symbol? Thus, we may view h_μ as the intrinsic unpredictability associated with the string; h_μ is the irreducible randomness in the spatial configurations, the randomness that persists as larger and larger blocks of spins are considered.

For a string generated by the tossing of a fair coin, the entropy rate is one bit per symbol. The coin tosses are independent; knowledge of the previous tosses tells you nothing about the outcome of the next toss. On the other hand, if we were considering a highly correlated process, the entropy rate would be much smaller. If there are strong correlations between symbols, knowledge of all the previous symbols will greatly decrease our uncertainty about the value of the next. The entropy rate captures the randomness or unpredictability inherent in the process.

There is yet another way to express the entropy density. It is not hard to show that:

$$h_\mu \;=\; \lim_{L\to\infty} \left[H(L+1) - H(L)\right] \;. \tag{2.8}$$

As we shall see in the next section, this way of writing the entropy density makes it clear that h_μ is the growth rate in the entropy as larger blocks of variables are considered.

Eqs. (1.7), (2.8), and (2.7) give different expressions for the entropy rate h_μ. These are all equivalent in the present setting, though they need not be for nonequilibrium or nonstationary processes.

2.1.1 Entropy Density and Kolmogorov-Chaitin Complexity

The entropy rate h_μ is related to the Kolmogorov-Chaitin (KC) complexity. The KC complexity of an object is the length of the minimal Universal Turing Machine (UTM) program needed to reproduce it. It turns out that h_μ is equal to the average length (per variable) of the minimal program that, when run, will cause a universal Turing machine to produce a typical configuration and then halt [8, 32].

This result is not that surprising. We saw in Chapter 1 that the Shannon entropy of a variable is equal to the average length of the optimal binary encoding for that variable. In this sense, $H[x]$ provides a measure of the average length of description of X—although a description of a particular form: binary coding. KC complexity measures a different type of description length: input programs for a UTM. There are certainly big differences between binary encoding and programs to be given to a UTM. However, in the limit that we are encoding an infinitely long string, these differences don't matter; both the UTM program and the binary encoding will *grow* at the same rate.

2.1.2 What Entropy Density Isn't

Let's conclude this section with an example. Consider the following two strings:

$$\overleftrightarrow{S}_A = \ldots 10101010101010101010101010101010 \ldots \ , \tag{2.9}$$

and

$$\overleftrightarrow{S}_B = \ldots 10101100101011001010110010101100 \ldots \ . \tag{2.10}$$

Both of these strings are periodic; a given block of symbols repeats indefinitely. As such, both can be predicted with certainty and both have zero entropy density; to predict the values of successive symbols, all one has to do is remember where in the pattern one is. But clearly these two systems aren't the same. The period of $\overleftrightarrow{S}_B$'s pattern is longer than that of $\overleftrightarrow{S}_A$; thus one might expect that in some sense $\overleftrightarrow{S}_B$ is "harder" to predict than $\overleftrightarrow{S}_A$. This is a distinction between systems that h_μ does not make. The entropy density indicates how predictable a system is—it says nothing about how hard it is to do the predicting. How can we measure this feature that the entropy density misses? Stay tuned.

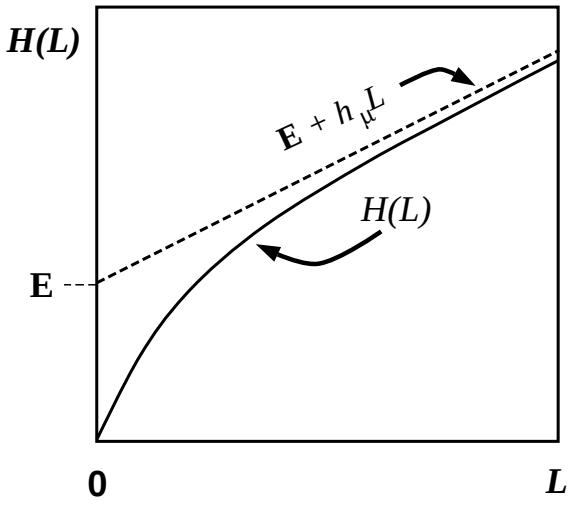

Figure 2.1: Total thermodynamic entropy growth: a schematic plot of $H(L)$ versus L. $H(L)$ increases monotonically and asymptotes to the line $\mathbf{E} + h_\mu L$, where $\mathbf{E}$ is the excess entropy, and h_μ is the thermodynamic entropy density.

2.2 Entropy Growth and Convergence

The Shannon entropy $H(L)$ over L-blocks is a monotonic increasing function of L. This is a simple consequence of the equality $H(L+1) \geq H(L)$ [8]. A schematic plot of $H(L)$ vs. L is shown in Fig. (2.1).

Recall that Eq. (2.8) showed us that the entropy density can be written as the difference between $H(L+1)$ and $H(L)$ in the limit that L goes to infinity. As as result, we see that the "terminal velocity" (*i.e.*, the slope as $L \to \infty$) of the curve in Fig. (2.1) corresponds to the entropy density h_μ.

The entropy density is a property of the system as a whole; only in special cases will the isolated-spin uncertainty $H(1)$ be equal to h_μ. It is natural to ask, therefore, how random the chain of spins appears when finite-length spin blocks are considered. That is, how do finite-L approximations of the entropy density converge to h_μ? To help us answer these questions, we define the following quantity:

$$h_\mu(L) \equiv H(L) - H(L-1), \ L = 1, 2, \ldots \ , \tag{2.11}$$

the incremental increase in uncertainty in going from $(L-1)$-blocks to L-blocks. We define $H(0) \equiv 0$.

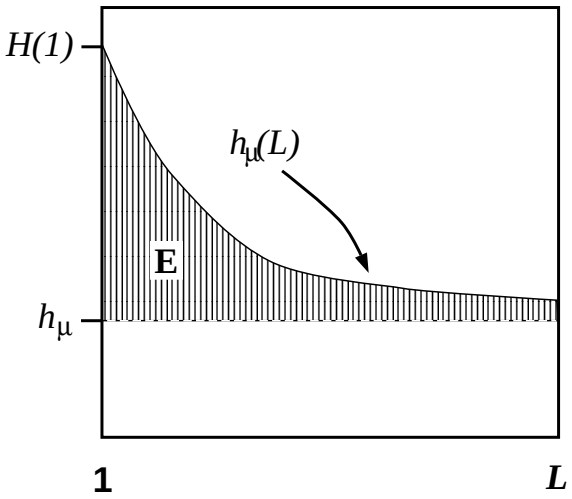

Figure 2.2: Entropy density convergence: A schematic plot of $h_\mu(L)$ versus L using the "typical" $H(L)$ shown above in Fig. 2.1. The entropy density h_μ asymptote is indicated by the horizontal dashed line. The shaded area is the excess entropy **E**.

Comparing Eq. (2.11) with Eq. (2.8), we see that $h_\mu(L)$ may be viewed as the finite-L approximation to the thermodynamic entropy density h_μ. Graphically, $h_\mu(L)$ is the two-point slope of the $H(L)$ vs. L curve. The convergence of $h_\mu(L)$ to h_μ is illustrated in Fig. (2.2). The entropy density h_μ is indicated by a horizontal dashed line.

The length-L approximation to the entropy density $h_\mu(L)$ overestimates the entropy density h_μ by an amount $h_\mu(L) - h_\mu$ that indicates how much more random the finite L-blocks appear than the infinite configuration $\overleftrightarrow{S}$. In other words, this excess randomness tells us how much additional information must be gained about the configurations in order to reveal the actual per-spin uncertainty h_μ. Summing up the overestimates one obtains the total excess entropy [15, 47, 45, 24, 35, 34, 33]

$$\mathbf{E} \equiv \sum_{L=1}^{\infty} [h_\mu(L) - h_\mu] \ . \tag{2.12}$$

Graphically, **E** is the shaded area in Fig. (2.2). If one inserts Eq. (2.11) into Eq. (2.12), the sum telescopes and one arrives at an alternate expression for

the excess entropy

$$\mathbf{E} = \lim_{L \to \infty} \left[H(L) \, - \, h_\mu L \right] \, . \tag{2.13}$$

Hence, $\mathbf{E}$ is the y-intercept of the straight line to which $H(L)$ asymptotes as indicated in Fig. (2.1).

Looking at Eq. (2.12), we see that, informally, $\mathbf{E}$ is the amount (in bits), above and beyond h_μ, of *apparent* randomness that is eventually "explained" by considering increasingly longer spin-blocks. Conversely, to see the actual (asymptotic) randomness at rate h_μ, we must extract $\mathbf{E}$ bits of information from measurements of spin blocks. Thus, we would expect a large $\mathbf{E}$ to indicate a large amount of structure; $\mathbf{E}$ is large if there are larger scale correlations which account for the apparent randomness observed when distributions over small blocks of spins are considered.

This interpretation is strengthened by noting that $\mathbf{E}$ may be expressed as the mutual information I, defined in Eq. (1.24), between the two semi-infinite halves of a configuration;

$$\mathbf{E} = I(\overleftarrow{S}; \overrightarrow{S}) \equiv \sum_{\{\overleftrightarrow{s}\}} \Pr(\overleftrightarrow{s}) \log_2 \left[\frac{\Pr(\overleftrightarrow{s})}{\Pr(\overleftarrow{s})\Pr(\overrightarrow{s})} \right] \, . \tag{2.14}$$

Note that this form makes it clear that $\mathbf{E}$ is spatially symmetric. The mutual information can also be written as the difference between a joint and conditional entropy [8]:

$$I(\overleftarrow{S}; \overrightarrow{S}) \, = \, H[\overleftarrow{S}] - H[\overleftarrow{S} \mid \overrightarrow{S}] \, . \tag{2.15}$$

In other words, $\mathbf{E}$ measures the average reduction in uncertainty of $\overleftarrow{S}$, given knowledge of $\overrightarrow{S}$. One must carefully view Eq. (2.14) since it contains entropy components, like $H(\overleftrightarrow{S})$, that may be individually infinite—even for a fair coin process.

Eqs. (2.14) and (2.15) allow us to interpret $\mathbf{E}$ as a measure of how much information one half of the spin chain carries about the other. In this restricted sense $\mathbf{E}$ measures the spin system's apparent spatial memory. If the configurations are perfectly random or periodic with period 1, then $\mathbf{E}$ vanishes. Excess entropy is nonzero between the two extremes of ideal randomness and trivial predictability, a property that ultimately derives from its expression as a mutual information. That is, the mutual information between two variables vanishes either (i) when the variables are statistically independent or (ii) they have no entropy or information to share. These

extremes correspond to **E** vanishing in the cases of ideal randomness and trivial predictability, respectively.

To summarize, then, the excess entropy **E** provides a measure of the apparent memory stored in a spatial configuration. Colloquially, **E** tells us how much the left half of the configuration "remembers" about the left. Another way of viewing this is that **E** is the "cost of amnesia"—the excess entropy measures how much more random the system would become if we suddenly forgot all information about the left half of the string.

2.3 History of Excess Entropy

The total excess entropy was first introduced by Crutchfield and Packard in refs. [15, 14, 13, 37] where they examined entropy convergence for noisy discrete-time nonlinear mappings. They developed a scaling theory for the entropy convergence rate γ: $h_\mu(L) - h_\mu \propto 2^{-\gamma L}$, where, for Markovian finite-memory discrete-time sources, the excess entropy and entropy convergence are simply related: $\mathbf{E} = (H(1) - h_\mu)/(1 - 2^{-\gamma})$. Analytical calculations of entropy convergence or **E** for some simple discrete-time nonlinear maps were carried out by Szépfalusy and Györgyi [47]. Excess entropy was recoined "stored information" by Shaw [45] and subsequently "effective measure complexity" by Grassberger [24]. These two authors emphasize the view shown in Fig. 2.1. It has been discussed in the context of cellular automata by Grassberger [24] and by Lindgren and Nordahl [35]. Excess entropy is also mentioned briefly by Lindgren in ref. [34]. The quantity is simply called "complexity" when applied to simple stochastic automata by Li [33]. Crutchfield and I have calculated the excess entropy for one-dimensional spin systems with finite-range interactions [11] and have compared the excess entropy to existing statistical mechanical measures of structure and correlation [22]. We also discuss the excess entropy in [22]. As noted above, Refs. [18, 20, 49] are recent reviews of entropy convergence and excess entropy.

Refs [24, 35] both provide fairly readable introductions to excess entropy. (These references also serve as a reminder that the study of "complexity" is not a phenomena that began in the '90's!) Ref. [22] is also intended to be a clear introduction to excess entropy and statistical complexity. These lecture notes are an expanded version of the introductory sections of ref. [22]. Ref. [45] is also recommended.

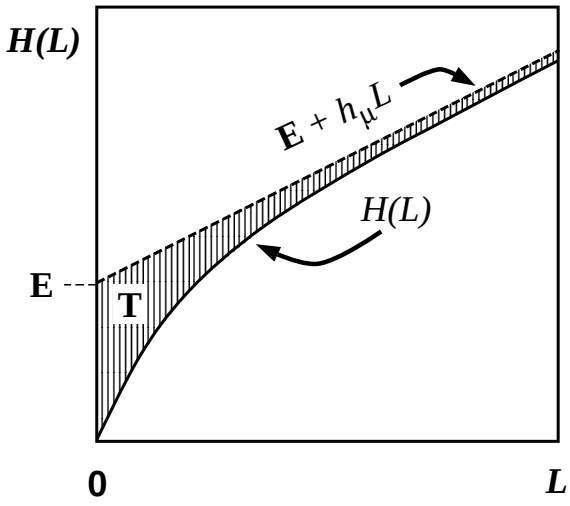

Figure 2.3: Total thermodynamic entropy growth: a schematic plot of $H(L)$ versus L. $H(L)$ increases monotonically and asymptotes to the line $\mathbf{E} + h_\mu L$, where $\mathbf{E}$ is the excess entropy, and h_μ is the thermodynamic entropy density. The shaded area is the transient information $\mathbf{T}$.

2.4 Transient Information

Finally, I mention briefly a new information-theoretic measure of structure, the *transient information*, introduced by Crutchfield and myself in Ref. [18] and discussed further in Refs. [12] and [21].

The transient information $\mathbf{T}$ measures the manner in which the total block entropy $H(L)$ approaches its asymptotic value $\mathbf{E} + h_\mu L$. Specifically, it is defined by:

$$\mathbf{T} \equiv \sum_{L=0}^{\infty} \mathbf{E} + h_\mu L - H(L) \ . \tag{2.16}$$

Graphically, the transient information is the shaded area in Fig.2.3.

As discussed in Refs. [18, 12], the transient information is a measure of how difficult it is to synchronize to an information source. If the source is Markovian, and we picture a scenario in which the observer has an accurate model of the process's internal states, then the transient information is related to the total internal-state-uncertainty experienced by an observer while synchronizing.

In Ref. [21], we report the results of exhaustively calculating the transient

information for all distinct periodic sequences up to and including period 23. This allows us to make a number of observations about the different structural properties of different sequences with the same period. These observations cannot be made by using the excess entropy, since the excess entropy for any sequence of period P is $\log_2 P$.

For a much more thorough discussion of the transient information and its applications and implications, the reader is referred to Refs. [18, 12, 21].

Chapter 3

Computational Mechanics

Note: This section is considerably out of date and is probably less polished and certainly less thoroughly referenced than the previous two chapters. A recent mathematical review of computational mechanics is found in Ref. [41]. Links to tutorials and pedagogical pieces on computational mechanics can be found at `http://www.santafe.edu/projects/CompMech/tutorials/CompMechTutorials.html`.

In the previous chapter we saw that the excess entropy **E** provides a measure of the spatial memory stored in configurations. However, we cannot interpret this as the memory needed to statistically reproduce the configurations, although we shall see in section 3.4.5 that these two subtly different notions of memory aren't unrelated. More importantly, excess entropy and the apparatus of information theory tell us nothing about *how* the system's memory is utilized. Computational mechanics addresses this issue, by making use of the architectural models of computation theory. For a review of computation theory, see, for example, refs. [7, 29]. The tools and ideas of computational mechanics have to date only appeared in research literature. A brief summary of references can be found in sec. (3.5). We shall see that this additional set of theoretical tools will allow us to describe structure and information processing at a more specific and complete level than we can by relying on information theory alone.

The basic motivating questions of computational mechanics concern how a system processes information. That is, in a system of many components, how is information stored, transmitted, and transformed? For example, how much information does one half of a spin configuration carry about the other? How much memory is needed to statistically reproduce an ensemble

of configurations? In general, we are interested in inferring the intrinsic computation being performed by the system itself.

By *intrinsic* computation we mean something very different than "computation" as the word is typically used in reference either to the use of modern digital computers as tools for simulation (e.g. "computational physics") or to the use of a device to perform useful information processing for some person, like the updating of a spreadsheet or determining the five billionth digit of π. Useful computation usually entails fixing the initial conditions and/or control parameters of a dynamical system so that the outcome contains some information of interest to us, as outside interpreters of the result. For example, we might employ the mapping

$$x_{n+1} = \frac{1}{2}(x_n + \frac{a}{x_n}) \, , \tag{3.1}$$

which has the useful property that $\lim_{n\to\infty} x_n = \sqrt{a}$ [31]. This iterative procedure for increasingly accurate estimates of roots is reported by Hero of Alexandria [36].

In contrast, when we ask about intrinsic computation, we are interested not in manipulating a system to produce an output that is useful to us—which is akin to an engineering stance towards nature. Instead, we are interested in examining the information processing that the system itself performs and the underlying mechanisms that support it. As a concrete example, consider the two-dimensional nearest-neighbor Ising model at the critical temperature. Here the correlations between spins decay with a power law as a function of distance, yet the total magnetization of the system remains zero. Computational mechanics is concerned with what sorts of effective computation the system must perform to reach and maintain the critical state. How much historical and/or spatial memory is required? Are the critical configurations in any way "harder" to reach than the low or high temperature behavior? More informally, how does the system balance up and down spins so that the correlations decay as a power law, while keeping zero magnetization?

3.1 Causal States and ϵ-machines: Preliminary Examples

Rather than launching into a flurry of mathematical definitions, we begin our review of computational mechanics by considering several simple examples.

After considering these examples, we shall see that we are led quite naturally to the definitions put forth in the following section.

The questions we shall be addressing are: how can one statistically reproduce a given bi-infinite configuration using the minimal amount of memory? In particular, how much information about the left half must be remembered to produce the right half? And what must we do with this information? Another, equivalent way of stating these questions is: How much memory is needed to optimally predict configurations? And how is this memory to be used? Optimal prediction corresponds to being able to predict the value of the next variable well enough so that the entropy associated with the prediction equals h_μ, the entropy density of the system.

3.1.1 Example 1: A Fair Coin

Consider a string generated by a fair coin toss:

$$\overleftrightarrow{S}_\alpha \equiv \ldots THTTTHHHHHTHTHTTHHT \ldots \quad . \tag{3.2}$$

All the symbols are independently distributed and the probability that any particular symbol is a heads is 1/2. We begin by asking: How much of the left half is needed to predict the values of the right half? Restated, imagine walking down the configuration from left to right, making a note of the variables you observe as you cross them. After having walked along the chain of variables for a very long time—long enough for you to have observed as many spins as you wish—how many spin variables must you keep track of so that you can optimally predict the spins you will encounter later in your left to right journey?

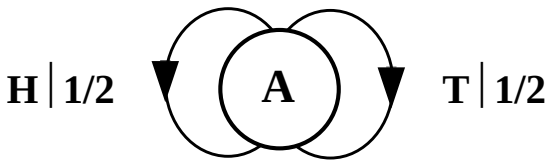

Figure 3.1: The ϵ-machine for a fair coin toss. This machine is a model of the original configuration in the sense that a random walk through the machine will statistically reproduce the configuration. For more discussion, see text.

A moment's thought indicates one does not need to keep track of any variables. Since the coin tosses are independent, knowledge of previous

tosses does not reduce your uncertainty about the next toss. As a result, for this particularly simple example no memory is required to optimally predict subsequent variables. Here, optimal prediction isn't that good—the entropy of the next coin toss is 1, a manifestation of the fact that the entropy density h_μ of the coin toss is 1.

What must one do to perform this optimal prediction? Equivalently, how can one statistically reproduce the configuration? The answer to these questions is illustrated in fig. (3.1). Borrowing from the computer science lexicon, the mathematical entity of fig. (3.1) is called an *ϵ-machine*. (The reason for the ϵ will be explained below.) The ϵ-machine of fig. (3.1) tells us how to statistically reproduce the original configuration generated by the coin toss. The machine is operated as follows: Start in state A. With probability 1/2, generate a H and return to state A. With probability 1/2, generate a T and also return to state A. A random walk through the machine following these rules results in a string of H's and T's that is statistically identical to $\overleftrightarrow{S}_\alpha$. In this sense we say that the ϵ-machine constitutes a model of the original process $\overleftrightarrow{S}_\alpha$.

3.1.2 Example 2: Period 1 Configuration

Let's now consider a string consisting of all 1's:

$$\overleftrightarrow{S}_\beta \equiv \ldots 1111111111111111111 \ldots \quad . \tag{3.3}$$

As with the fair coin, it is clear that one doesn't need to remember any of the previous symbols to perform optimal prediction. The value of the next variable will be 1 no matter what the previous variables were. The value of the next variable can be predicted with probability 1, as reflected by the zero entropy density h_μ.

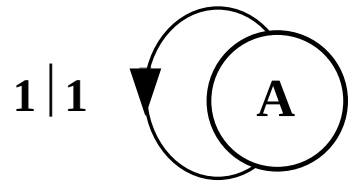

Figure 3.2: The ϵ-machine for a string consisting of all 1's

The ϵ-machine for $\overleftrightarrow{S}_\beta$ is shown in fig. (3.2). From state A, the machine always outputs the symbol 1 and returns to state A. In this way the machine

statistically reproduces $\overleftrightarrow{S}_\beta$. For this example the reproduction is exact, since $h_\mu = 0$.

3.1.3 Example 3: Period 2 Configuration

As a final example, we consider an infinite, period 2 configuration:

$$\overleftrightarrow{S}_\gamma \cdots \uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow \cdots . \tag{3.4}$$

Again, we begin by asking: How much of the left half is needed to predict the values of the right half? This time, some memory is needed. One will need to keep track of one spin, corresponding to the phase of the pattern. Once this spin value is known you can optimally predict all the subsequent spins. This prediction can be made with certainty since the entropy density of $\overleftrightarrow{S}_\gamma$ is zero. To perform this prediction the values of spins, one must distinguish between the two different phases of the pattern. As a result, the ϵ-machine for $\overleftrightarrow{S}_\gamma$ has two states, as indicated in fig. (3.3)

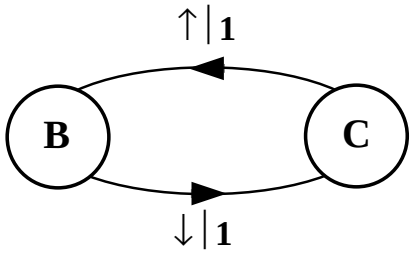

Figure 3.3: The recurrent portion of the ϵ-machine for the period 2 configuration, Eq. (3.4).

How can we use the machine of fig. (3.3) to reproduce $\overleftrightarrow{S}_\gamma$? Unlike our previous examples, it is not clear where to begin: B or C? A first response, in keeping with the statistical mechanics tradition of considering mainly equilibrium, infinite systems, is that it doesn't matter. If we run the system for infinitely long we will statistically reproduce the original configuration.

However, in another sense the state in which we start most definitely *does* matter. Suppose we always choose to start in state B. We then examine all the length 3 strings output by this model. We see that the string $\uparrow\downarrow\uparrow$ is

generated each time. Yet in the original configuration, Eq. (3.4), we observe $\Pr(\uparrow\downarrow\uparrow) = 1/2$ and $\Pr(\downarrow\uparrow\downarrow) = 1/2$. Our model doesn't get the statistics of the configuration right if it outputs finite length strings.

There is an easy remedy for this situation: start in A half the time and B half the time. We can achieve this by adding a *start state* to our model, as shown in Fig. (3.4). We now always begin operating our model in the unique start state A. In Fig (3.4) and all subsequent figures the start state will be indicated with a double circle. We can use our new, improved model to generate finite-length strings that faithfully reproduce the distribution of finite length spin blocks observed in the original configuration.

The start state is a *transient state*; it is never revisited after the machine outputs a symbol and moves to state B or C. The states B and C in fig. (3.4) are recurrent states, being visited infinitely often (with probability 1) as the model is operated and an infinite string of symbols is produced. In general determining how to begin operating the machine will not always be a simple as choosing one of the recurrent states at random, as was the case for this particular example

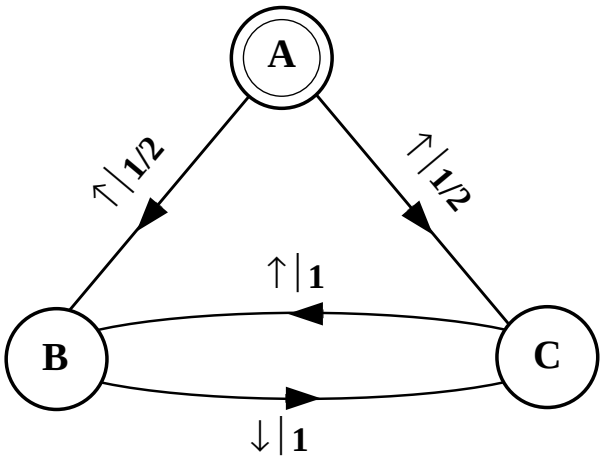

Figure 3.4: The full ϵ-machine for the period 2 example. The start state, A, is indicated by the double circle. A is a transient state which is never visited again after the machine outputs the first symbol. States B and C are recurrent, visited infinitely often as the machine outputs an infinite string of symbols.

3.1.4 Summary of Examples

A few summarizing remarks are in order before moving on to state the mathematical definition of an ϵ-machine. First note that the coin toss $\overleftrightarrow{S}_\alpha$ and the period 1 configuration $\overleftrightarrow{S}_\beta$ both result in an ϵ-machine with only one state, an indication that we don't need to remember any information about the previous spins to predict the values of the next. Thus, we see that predicting a perfectly random process and a process with a very simple configuration are both "easy" tasks in the sense that they require a machine with only one state.

Second, note that h_μ manifests itself as branching in the ϵ-machine. An example of branching is shown in fig. (3.1); there are two arrows leaving state A. Lastly, note that the structure of the ϵ-machine does not depend on the names of the variables—all that matters is the probabilities over configurations. For example, if the symbols H and T are changed to $\uparrow$ and $\downarrow$, the ϵ-machine of fig. (3.1) will output different symbols, but its overall structure remains unchanged.

3.2 Definitions of Causal States and ϵ-machines

In the preceding section we generated a model, illustrated in fig. (3.4), that is capable of reproducing the distribution of finite and infinite length blocks of spins observed in the original translationally invariant infinite configuration. In this section we put forth a general procedure for constructing such a model.

First, we seek to generalize the process through which the "effective" states of the three example systems were discovered. The key step is to identify the notion of state with the conditional probability distribution over right half configurations. When forming a model, there is no need to distinguish between different left half configurations that give rise to an identical state of knowledge about the right half configurations that can follow it. Maintaining a distinction between two such states adds to the memory requirements of the model without increasing its predictive ability.

To make this idea precise, consider the probability distribution of all possible right halves $\overrightarrow{s}$ conditioned on a particular left half $\overleftarrow{s}_i^L$ of length L at site i: $\Pr(\overrightarrow{s} \mid \overleftarrow{s}_i^L)$. Here, $0 < L < \infty$; for $L = 0$, $\overleftarrow{s}_i^L$ is the empty string, denoted by λ. That is, $\Pr(\overrightarrow{s} \mid \overleftarrow{s}_i^0) \equiv \Pr(\overrightarrow{s} \mid \lambda) = \Pr(\overrightarrow{s})$ denotes the probability of observing $\overrightarrow{s}$ unconditioned on any spins in the left half of the configuration.

We now use this form of conditional probabilities to define an equivalence relation $\sim$ on the space of all left halves; the induced equivalence classes are subsets of the set of all allowed $\overleftarrow{s}_i^L$. We say that two configurations at different lattice sites are equivalent (under $\sim$) if and only if they give rise to conditional distributions over right-half configurations that are identical up to some tolerance δ. Formally, we define the relation $\sim$ by

$$\overleftarrow{s}_i^L \sim \overleftarrow{s}_j^L \text{ iff } \Pr(\overrightarrow{\mathrm{s}} \,|\overleftarrow{\mathrm{s}}_{\mathrm{i}}^{\mathrm{L}}) = \Pr(\overrightarrow{\mathrm{s}} \,|\overleftarrow{\mathrm{s}}_{\mathrm{j}}^{\mathrm{L}}) + \delta \;\; \forall \; \overrightarrow{\mathrm{s}} \;, \tag{3.5}$$

where δ is a tolerance. In the discussion to follow δ is effectively set to zero. Thus, we require exact equality; $\Pr(\overrightarrow{\mathrm{s}} \,|\overleftarrow{\mathrm{s}}_{\mathrm{i}}^{\mathrm{L}}) = \Pr(\overrightarrow{\mathrm{s}} \,|\overleftarrow{\mathrm{s}}_{\mathrm{j}}^{\mathrm{L}})$.

Note that there is a mirror image definition of causal states that correspond to scanning the lattice in the opposite direction. Finite-memory Markov chains respect this symmetry, so for this restricted class of systems the causal states will be the same regardless of the scanning direction. In the general case, in which this reversal symmetry need not hold, it it possible to find different causal states if one scans $\overleftrightarrow{S}$ in different directions.

The equivalence classes induced by this relation are called *causal states* and denoted $\mathcal{S}_i$. These are the "effective states" of the process referred to above. Two $\overleftarrow{s}^L$ belong to same causal state if, as measured by the probability distribution of subsequent spins conditioned on having seen that particular left-half configuration, they give rise to the same degree of certainty, within δ, about the configurations that follow to the right. The equivalence class that contains $\Pr(\overrightarrow{s} \,|\, \lambda)$ is always the start state, as this distribution corresponds to the distribution known before any spins are observed.

As we saw above, for the period-2 system there are 3 causal states, indicated in Fig. (3.4) by A, B, and C. These causal states are subsets of the allowed $\overleftarrow{s}^L$;

$$A = \{\lambda\} \;, \tag{3.6}$$

$$\begin{aligned} B &= \{\,\overleftarrow{s}^L | s_{-1} = \downarrow, s_i = s_{i+2}, L \geq 1\} \\ &= \{\downarrow, \uparrow\downarrow, \downarrow\uparrow\downarrow, \uparrow\downarrow\uparrow\downarrow, \downarrow\uparrow\downarrow\uparrow\downarrow, \ldots\} \;, \end{aligned} \tag{3.7}$$

and

$$\begin{aligned} C &= \{\,\overleftarrow{s}^L | s_{-1} = \uparrow, s_i = s_{i+2}, L \geq 1\} \\ &= \{\uparrow, \downarrow\uparrow, \uparrow\downarrow\uparrow, \downarrow\uparrow\downarrow\uparrow, \uparrow\downarrow\uparrow\downarrow\uparrow, \ldots\} \;, \end{aligned} \tag{3.8}$$

The causal states, as determined by the equivalence classes induced by Eq. (3.5), give transient as well as recurrent states. Defined more carefully

than above, transient states are those causal states that are visited infinitely often with probability 0 in the limit that the machine produces an infinite configuration. Recurrent states are those visited infinitely often with probability 1 in the same limit. If one is only interested in the recurrent states, one need only determine the equivalence classes obtained when the $L \to \infty$ limit is considered in Eq. (3.5).

We denote the set of causal states by $\mathcal{S} = \{\mathcal{S}_i, i = 1, \ldots, k\}$, where for Markovian processes $\mathcal{S}$ is discrete and k is finite—neither of which need to be true in a more general setting [9, 48].

For the period-2 example, $\mathcal{S} = \{A, B, C\}$. Let $\mathcal{S}^{(T)}$ denote the set of transient states and $\mathcal{S}^{(R)}$ denote the set of recurrent states. For the period-2 example $\mathcal{S}^{(T)} = \{A\}$ and $\mathcal{S}^{(R)} = \{B, C\}$. Note that $\mathcal{S} = \mathcal{S}^{(T)} \cup \mathcal{S}^{(R)}$.

Once the set of causal states $\mathcal{S}$ has been identified, we determine the transition probabilities $T_{ij}^{(s)}$ between states upon seeing symbol $s \in \mathcal{A}$. $T = \sum_{s \in \mathcal{A}} T^{(s)}$ is a matrix whose components T_{ij} give the probability of a transition from the i^{th} to the j^{th} causal state;

$$T_{ij} \equiv \Pr(\mathcal{S}_j | \mathcal{S}i) \ . \tag{3.9}$$

Since the probabilities are normalized, $\sum_j T_{ij} = 1$ and T is a stochastic matrix—the probability of leaving a state is unity. Thus, $\Pr(\mathcal{S}_i)$, the probability of finding the chain in the i^{th} causal state after the machine has been running infinitely long is given by the left eigenvector of T with eigenvalue 1, normalized in probability. That is, $\Pr(\mathcal{S}_i)$ is given by:

$$\sum_{i=1}^{\|\mathcal{S}\|} \Pr(\mathcal{S}_i) T_{ij} \ = \ \Pr(\mathcal{S}_j) \ . \tag{3.10}$$

The asymptotic probability of all transient states is zero;

$$\Pr(\mathcal{S}_i) \ = \ 0 \quad \forall \, \mathcal{S}_i \in \mathcal{S}^{(T)} \ . \tag{3.11}$$

The set $\{\mathcal{S}_i\}$ together with the dynamic T constitute a model—referred to as an ϵ-machine [16]—of the original infinite configurations. The ϵ-machine is a minimal representation of the intrinsic computation being performed by the system under study. The "ϵ" signifies that, in general, the measurements may not be direct indicators of the internal states. For example, the symbols may be discretizations of measurements that are continuous in space and/or time.

Note that the determination of an ϵ-machine does *not* depend on knowledge of the dynamics or rule through which the configurations were generated. The causal states and their transition probabilities may be calculated

given access to the configurations themselves. This procedure through which this is done is referred to as *ϵ-machine reconstruction*.

3.3 What do ϵ-machines represent?

The ϵ-machines so defined are a special class of deterministic finite state machines [7, 29] that have the following properties: (i) a unique start state, (ii) all states are accepting, (iii) all recurrent states form a single strongly connected component in the machine's state graph. Finally, unlike finite state machines ϵ-machines transitions are labeled with conditional probabilities. ϵ-machines can also be viewed as a type of Markov chain. More correctly they are called "functions of Markov chains" or hidden Markov models, since the output alphabet differs from the internal state alphabet [6].

An essential feature of computational mechanics is that it begins by trying to model the original process using the *least* powerful model class. That is, simple finite-memory machines are employed first. However, as noted above, finite-memory machines may fail to admit a finite size model—the number of causal states could turn out to be infinite. If this is the case, a model more powerful than a deterministic finite state machine must be used. One proceeds by trying to use the next most powerful model in a hierarchy of machines known as the causal hierarchy [9], in analogy with the Chomsky hierarchy of formal language theory [7, 29].

The ϵ-machine provides a minimal description of the pattern or regularities in a system in the sense that the pattern *is* the algebraic structure determined by the causal states and their transitions [51]. If, for example, the ϵ-machine has an algebraic structure that is a group, then it captures a translation symmetry is pattern of the configurations "pattern." Typically, though, the algebraic structure is a semi-group and so not so easily interpreted in terms of "symmetries." Nonetheless, the algebraic structure is still the "pattern."

The ϵ-machine is a model of the original configuration. From this model, we can proceed to define and calculate macroscopic or global properties that reflect the characteristic average information processing capabilities of the system. This will be the subject of the following few sections.

3.4 Global Properties from ϵ-machines

3.4.1 ϵ-Machine Entropy Rate

Recall that we saw in Eq. (2.7) that the entropy density h_μ can be expressed as the conditional entropy of one spin conditioned on all those spins that came before it. Using this, it is not hard to show that the entropy density can be reexpressed in terms of the distribution over the causal states:

$$h_\mu = -\sum_{\{\mathcal{S}_i\}} \sum_{s \in \mathcal{A}} \Pr(s, \mathcal{S}_i) \log_2 \Pr(s|\mathcal{S}_i) \; . \tag{3.12}$$

This result, derived carefully in appendix B, is not that surprising given the definition of causal states. In defining the causal states, configurations that led to the same conditional distribution over possible right half configurations were grouped together. As a result, to calculate the entropy density, one only need consider the conditional entropy of a single spin conditioned on the previous causal states.

3.4.2 ϵ-Machine excess entropy

The excess entropy $\mathbf{E}$ can also be calculated from the probabilities of the causal states and their transitions. In the most general setting there is no compact formula for $\mathbf{E}$ in terms of $\Pr(\mathcal{S})$ and $\Pr(s|\mathcal{S})$, as there was for h_μ. However, for the special case where the causal states are in a one-to-one correspondence with the values of blocks of the observed variables S, it is possible to write down a relatively simple formula for $\mathbf{E}$ in terms of an ϵ-machine. An example of this is given in [22], where ϵ-machines and the excess entropy are calculated for one-dimensional Ising systems with finite range interactions.

3.4.3 Statistical Complexity

In the previous section, we saw how to calculate the entropy density and the excess entropy from the ϵ-machine. Motivated by the question: how much memory is needed to operate this machine? — we now define a new quantity.

To predict the successive spins in a configuration with an ϵ-machine as one scans from left to right, one must track in which causal state the process is, since knowledge of the causal state gives the appropriate conditional distribution. Thus, the informational size of the distribution over causal states $\Pr(\mathcal{S}_i)$, as measured by the Shannon entropy, gives the minimum average

amount of memory needed to optimally predict the right-half configurations. This quantity is the *statistical complexity* [16];

$$C_\mu \equiv -\sum_{\{\mathcal{S}_i\}} \Pr(\mathcal{S}_i) \log_2 \Pr(\mathcal{S}_i) \ . \tag{3.13}$$

Another, coarser measure of the ϵ-machine's size is simply the number of causal states. This motivates the definition of the *topological complexity* C_0 as the logarithm of the number of causal states [9]; that is,

$$C_0 \ = \ \log_2 \|\mathcal{S}\| \ . \tag{3.14}$$

The topological complexity ignores the probability of the sequences, simply describing which sequences occur and which do not.

3.4.4 ϵ-Machine Thermodynamics

ϵ-machines also provide a direct way to calculate the fluctuation spectrum, also known as the spectrum of singularities, "S of U curves" or "f of alpha" curves [25, 3]. The basic idea is to start with the matrix that gives the probabilities of transition between causal states as defined in Eq. (3.9). Each element of the matrix is then raised to the $\tilde{\beta}$ power:

$$T(\tilde{\beta})_{ij} \ \equiv \ (\Pr(\mathcal{S}_j|\mathcal{S}i))^{\tilde{\beta}} \ . \tag{3.15}$$

The parameter $\tilde{\beta}$ is used to scan different "regions" of the probability distribution. For $\tilde{\beta} = \infty$ only the most probable state is considered, corresponding to the ground state of the system. At $\tilde{\beta} = 0$ all configurations which occur with nonzero probability are weighted equally. Note that while $\tilde{\beta}$ acts like the inverse thermodynamic temperature, it is not identical to it. From this parameterized transition matrix $T(\tilde{\beta})_{ij}$ one can efficiently calculate the fluctuation spectrum. Details are given in Ref. [52].

In Ref. [52] it was shown that calculating the fluctuation spectrum by first determining the ϵ-machine and then proceeding as sketched above yields significantly more accurate results than calculating the spectrum directly from the configuration by using histograms to estimate probabilities.

3.4.5 Relationships between Quantities

It turns out that the excess entropy sets a lower bound on the statistical complexity:

$$\mathbf{E} \leq C_\mu \ . \tag{3.16}$$

This result is quite general; it holds for any translationally invariant infinite configuration [17]. Thus, the memory needed to perform optimal prediction of the right-half configurations cannot be lower than the mutual information between the left and right halves themselves. This relationship reflects the fact that the set of causal states is not in one-to-one correspondence with L-block or even ∞-length configurations. The causal states are a reconstruction of the hidden, effective states of the process.

For the special case of Markov chains (equivalently, finite-range spin systems), in which the values of R-blocks of the observed $S \in \mathcal{A}$ are in a one-to-one correspondence with the internal state alphabet $\mathcal{S}$, there is a precise relationship between C_μ, h_μ and $\mathbf{E}$:

$$C_\mu = \mathbf{E} + Rh_\mu \ . \tag{3.17}$$

This result is proved and thoroughly discussed in [22].

$$H[\mathcal{S}'|\mathcal{S}] \ = \ H[S|\mathcal{S}] \ = \ h_\mu \ . \tag{3.18}$$

The last equality follows from Eq. (3.12). In light of this, eq. (3.17) follows immediately.

3.4.6 Related, or not, Measures of "Complexity"

As noted above, an ϵ-machine is a model of the original process using the least powerful machine that admits a finite model. In sharp contrast, Kolmogorov-Chaitin (KC) complexity characterizes symbol sequences by considering their representation in terms of the most powerful of the computational models, universal Turing machines. Note that for both C_μ and $\mathbf{E}$ no memory is expended trying to account for the randomness or, in this case, for the thermal fluctuations present in the system. Thus, these measures of structural complexity depart markedly from the deterministic KC complexity. As noted above, the per-spin KC complexity is h_μ [8, 32].

A quantity more closely related to statistical complexity and excess entropy is the *logical depth* of Bennett [4]. Whereas the Kolmogorov-Chaitin complexity of a symbol string is defined as the length of the shortest universal Turing machine program capable of exactly reproducing that string, the logical depth is defined as the run time needed to implement the algorithm. If a string $\overleftrightarrow{S}_\alpha$ is random, the shortest UTM program that reproduces it is the program "Print($\overleftrightarrow{S}_\alpha$)." This is a very long program but takes very little time to run; the program contains only one command. On the other hand, if a string has a very simple pattern, say all 1's, then the program to reproduce

it also takes a quick time to run. All the machine needs to do is loop over the command "Print (1)." If the string has a great deal of structure, for example the binary expansion of π, then the minimal program to reproduce it will involve many operations, and hence take a long time to run.

As a result, like excess entropy and statistical complexity, the logical depth captures a property distinct from randomness and from that described by Kolmogorov-Chaitin complexity. Note, however, that C_μ is a measure of memory while logical depth is a measure of run time. A shortcoming of logical depth, which it shares with KC complexity, is that it is in general uncomputable [8, 32]. That is, unlike statistical complexity and excess entropy, there exists no general algorithm for its calculation. It should be noted, however, that in special cases such as finite-state Markov chains, the average value of the growth rate of the Kolmogorov-Chaitin complexity can be calculated and is equal to the Shannon entropy rate h_μ of the process.

For other approaches to statistical complexity and correlational structure, see refs. [5, 23, 30, 1, 50].

3.5 Computational Mechanics References

For a more detailed discussion of the motivations and central issues that underlie computational mechanics, see [9, 10]. Computational mechanics has been successfully adapted and applied to the period-doubling and quasiperiodic routes to chaos [16, 17], one-dimensional cellular automata [27, 28], globally coupled maps [19], recurrent hidden Markov models [9, 48], and one-dimensional Ising models [11, 22]. Computational mechanics has also been proposed [39] as a useful tool with which to reexamine the learning paradox of developmental psychology that concerns the discovery of new patterns, not seen before [10].

Most of the papers by Crutchfield and the Computational Mechanics Group can be found at `http://www.santafe.edu/projects/CompMech/papers/CompMechCommun.html`. The dissertations of Cosma Shalizi [42], Karl Young [51] Jim Hanson [26] and Dan Upper [48] might also make good reading.

Bibliography

[1] R. Badii and A. Politi. *Complexity: Hierarchical structures and scaling in physics*. Cambridge University Press, Cambridge, 1997.

[2] Ralph Baierlein. *Atoms and information theory; An introduction to statistical mechanics*. W. H. Freeman, 1971.

[3] C. Beck and F. Schlögl. *Thermodynamics of Chaotic Systems*. Cambridge University Press, 1993.

[4] C. H. Bennett. On the nature and origin of complexity in discrete, homogeneous locally-interacting systems. *Found. Phys.*, 16:585–592, 1986.

[5] C. H. Bennett. How to define complexity in physics, and why. In W. H. Zurek, editor, *Complexity, Entropy, and the Physics of Information*, volume VIII of *Santa Fe Institute Studies in the Sciences of Complexity*, pages 137–148. Addison-Wesley, 1990.

[6] D. Blackwell and L. Koopmans. On the identifiability problem for functions of Markov chains. *Ann. Math. Statist.*, 28:1011–1015, 1957.

[7] J. G. Brookshear. *Theory of Computation: Formal Languages, Automata, and Complexity*. Benjamin/Cummings, 1989.

[8] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.

[9] J. P. Crutchfield. The calculi of emergence: Computation, dynamics, and induction. *Physica D*, 75:11–54, 1994.

[10] J. P. Crutchfield. Is anything ever new? Considering emergence. In G. Cowan, D. Pines, and D. Melzner, editors, *Complexity: Metaphors, Models, and Reality*, volume XIX of *Santa Fe Institute Studies in the*

Sciences of Complexity, pages 479–497, Reading, MA, 1994. Addison-Wesley.

[11] J. P. Crutchfield and D. P. Feldman. Statistical complexity of simple one-dimensional spin systems. *Phys. Rev. E*, 55(2):1239R–1243R, 1997.

[12] J. P. Crutchfield and D. P. Feldman. Synchronizing to the environment: Information theoretic constraints on agent learning. *Advances in Complex Systems*, 4:251–264, 2001.

[13] J. P. Crutchfield and N. H. Packard. Noise scaling of symbolic dynamics entropies. In H. Haken, editor, *Evolution of Order and Chaos*, pages 215–227, Berlin, 1982. Springer-Verlag.

[14] J. P. Crutchfield and N. H. Packard. Symbolic dynamics of one-dimensional maps: Entropies, finite precision, and noise. *Intl. J. Theo. Phys.*, 21:433–466, 1982.

[15] J. P. Crutchfield and N. H. Packard. Symbolic dynamics of noisy chaos. *Physica D*, 7:201–223, 1983.

[16] J. P. Crutchfield and K. Young. Inferring statistical complexity. *Phys. Rev. Lett.*, 63:105–108., 1989.

[17] J. P. Crutchfield and K. Young. Computation at the onset of chaos. In W. H. Zurek, editor, *Complexity, Entropy and the Physics of Information*, volume VIII of *Santa Fe Institute Studies in the Sciences of Compexity*, pages 223–269. Addison-Wesley, 1990.

[18] J.P. Crutchfield and D.P. Feldman. Regularities unseen, randomness observed: Levels of entropy convergence. *Chaos*, 2001. arXiv.org/abs/cond-mat/0102181. In Press.

[19] J. Delgado and R. V. Solé. Collective-induced computation. *Phys. Rev. E*, 55(3):2338–2344, 1997.

[20] W. Ebeling. Prediction and entropy of nonlinear dynamical systems and symbolic sequences with LRO. *Physica D*, 109:42–52, 1997.

[21] D. P. Feldman and J. P. Crutchfield. Synchronizing to a periodic signal: The transient information and synchronization time of periodic sequences. Submitted to Physical Review E. arXiv/nlin.AO/0208040.

[22] D. P. Feldman and J. P. Crutchfield. Measures of statistical complexity: Why? *Physics Letters A*, 238:244–252, 1998.

[23] M. Gell-Mann and S. Lloyd. Information measures, effective complexity, and total information. *Complexity*, 2(1):44–52, 1996.

[24] P. Grassberger. Toward a quantitative theory of self-generated complexity. *Intl. J. Theo. Phys.*, 25(9):907–938, 1986.

[25] T.C. Halsey, M. H. Jensen, L.P. Kadanoff, I. Procaccia, and B. I. Shraiman. Fractal measures and their singularities: The characterization of strange sets. *Phys. Rev. A*, 33:1141–1151, 1986.

[26] J. E. Hanson. *Computational Mechanics of Cellular Automata*. PhD thesis, University of California, Berkeley, 1993.

[27] J. E. Hanson and J. P. Crutchfield. The attractor-basin portrait of a cellular automaton. *J. Stat. Phys.*, 66:1415–1462, 1992.

[28] J. E. Hanson and J. P. Crutchfield. Computational mechanics of cellular automata: An example. *Physica D*, 103(1-4):169–189, 1997.

[29] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, 1979.

[30] B. A. Huberman and T. Hogg. Complexity and adaptation. *Physica D*, 22:376–384, 1986.

[31] W. R. Knorr. *The Ancient Tradition of Geometric Problems*. Birkhauser, Boston, 1986.

[32] M. Li and P. M. B. Vitanyi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer-Verlag, New York, 1993.

[33] W. Li. On the relationship between complexity and entropy for Markov chains and regular languages. *Complex Systems*, 5(4):381–399, 1991.

[34] K. Lindgren. Microscopic and macroscopic entropy. *Phys. Rev. A*, 38(9):4794–4798, 1988.

[35] K. Lindgren and M. G. Norhdal. Complexity measures and cellular automata. *Complex Systems*, 2(4):409–440, 1988.

[36] Hero of Alexandria. *Opera*, volume III: Metrica. B. G. Teubner, Leipzig, 1903.

[37] N. H. Packard. *Measurements of Chaos in the Presence of Noise*. PhD thesis, University of California, Santa Cruz, 1982.

[38] J. R. Pierce. *Symbols, Signals, and Noise*. Harper & Brothers, 1961.

[39] M. Raijmakers. *Epigensis in Neural Network Models of Cognitive Development: Bifurcations, More Powerful Structures, and Cognitive Concepts*. PhD thesis, Universiteit van Amsterdam, 1996.

[40] Harry S. Robertson. *Statistical Theromphysics*. Prentice Hall, 1993.

[41] C. R. Shalizi and J. P. Crutchfield. Computational mechanics: Pattern and prediction, structure and simplicity. *Journal Statistical Physics*, 104:819–881, 2001.

[42] C.R. Shalizi. *Causal Architecture, Complexity and Self-Organization for Time Series and Cellular Automata*. PhD thesis, University of Wisconsin at Madison, 2001.

[43] C. E. Shannon. A mathematical theory of communication. *Bell System Tech. J.*, 27:379–423, 1948. as reprinted in "The Mathematical Theory of Communication", C. E. Shannon and W. Weaver, University of Illinois Press, Champaign-Urbana (1963).

[44] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, 1963.

[45] R. Shaw. *The Dripping Faucet as a Model Chaotic System*. Aerial Press, Santa Cruz, California, 1984.

[46] N. J. A. Sloane and A. D. Wyner, editors. *C. E. Shannon: Collected Papers*. IEEE Press, 1993.

[47] P. Szépfalusy and G. Györgyi. Entropy decay as a measure of stochasticity in chaotic systems. *Phys. Rev. A*, 33(4):2852–2855, 1986.

[48] D. R. Upper. *Theory and Algorithms for Hidden Markov Models and Generalized Hidden Markov Models*. PhD thesis, University of California, Berkeley, 1997.

[49] I. Nemenman w. Bialeck and N. Tishby. Complexity through nonextensivity. *Physica A*, 302:89–99, 2001.

[50] B. Wackerbauer, A. Witt, H. Atmanspacher, J. Kurths, and H. Scheingraber. A comparative classification of complexity measures. *Chaos, Solitons & Fractals*, 4(1):133–173, 1994.

[51] K. Young. *The Grammar and Statistical Mechanics of Complex Physical Systems*. PhD thesis, University of California, Santa Cruz, 1991.

[52] K. Young and J. P. Crutchfield. Fluctuation spectroscopy. *Chaos, Solitons, and Fractals*, 4:5–39, 1993.

Appendix A

Some Mathematical Details

A.1 Equivalence of Formulae for Entropy Rate

Our goal is to show that equations 2.6 and 2.7 are equivalent. What follows might be slightly less than rigorous. A rigorous demonstration, complete with ϵ's and δ's can be found in [8], pages 64-5.

We begin with Eq. (2.6).

$$\begin{aligned} h_\mu &\equiv \lim_{L\to\infty} \frac{H(L)}{L} && \text{(A.1)} \\ &= \lim_{L\to\infty} \left(\frac{H(S_0 S_1 \ldots S_{L-1})}{L} \right) && \text{(A.2)} \\ &= \lim_{L\to\infty} \frac{-1}{L} \sum_{\{S_i\}} \Pr(\mathrm{S_0 S_1 \ldots S_{L-1}}) \log \left[\Pr(\mathrm{S_0 S_1 \ldots S_{L-1}}) \right] \ . && \text{(A.3)} \end{aligned}$$

The sum over $\{S_i\}$ indicates that the sum is to be performed over all the possible values of all the S_i's.

We now factor the joint probabilities inside the argument of the log into a large product of conditional probabilities:

$$\begin{aligned} h_\mu = \lim_{L\to\infty} \Bigg\{ \frac{-1}{L} \sum_{\{S_i\}} \Bigg[& \Pr(\mathrm{S_0 S_1 \ldots S_{L-1}}) \times \\ & \log\{\Pr(\mathrm{S_{L-1}|S_0 \ldots S_{L-2}}) \Pr(\mathrm{S_{L-2}|S_0 \ldots S_{L-3}}) \times \\ & \Pr(\mathrm{S_{L-3}|S_0 \ldots S_{L-4}}) \cdots \} \Bigg] \Bigg\} \ . \end{aligned} \qquad \text{(A.4)}$$

All in all, there will be L conditional probabilities in the argument of the logarithm. In the $L \to \infty$ limit, I claim that all of these conditional probabil-

ities are equivalent to $\Pr(\mathrm{S_L}|\mathrm{S_0}\ldots\mathrm{S_{L-1}})$. This slightly dubious observation enables me to write:

$$h_\mu = \lim_{L\to\infty}\left\{\frac{-1}{L}\sum_{\{S_i\}}\Big[L\Pr(\mathrm{S_0S_1}\ldots\mathrm{S_{L-1}})\times \log[\Pr(\mathrm{S_{L-1}}|\mathrm{S_0}\ldots\mathrm{S_{L-2}})]\Big]\right\}. \tag{A.5}$$

The L's cancel, and upon comparison with Eq. (1.20), we see that

$$h_\mu = \lim_{L\to\infty} H(S_{L-1}|S_0\cdots S_{L-2})\,. \tag{A.6}$$

This is Eq. (2.7), thus completing our task.

A.2 Equivalence of Expressions for Excess Entropy

I aim to show that Eqs. (2.14) and (2.13) are equivalent. I begin with Eq. (2.14):

$$E = \mathrm{MI}(\overrightarrow{S}\;;\;\overleftarrow{S})\,. \tag{A.7}$$

Using Eq. (1.24), I may write:

$$E = \sum_{\{S_i\}}\Pr(\overleftarrow{\mathrm{S}}\,,\,\overrightarrow{\mathrm{S}})\log[\frac{\Pr(\overleftarrow{\mathrm{S}}\,,\,\overrightarrow{\mathrm{S}})}{\Pr(\overleftarrow{\mathrm{S}})\Pr(\overrightarrow{\mathrm{S}})}] \tag{A.8}$$

Factoring the joint probability, I may write the argument of the logarithm as:

$$\frac{\Pr(\overleftarrow{\mathrm{S}}\,,\,\overrightarrow{\mathrm{S}})}{\Pr(\overleftarrow{\mathrm{S}})\Pr(\overrightarrow{\mathrm{S}})} = \frac{\Pr(\overleftarrow{\mathrm{S}})\Pr(\overrightarrow{\mathrm{S}}\mid\overleftarrow{\mathrm{S}})}{\Pr(\overleftarrow{\mathrm{S}})\Pr(\overrightarrow{\mathrm{S}})} = \frac{\Pr(\overrightarrow{\mathrm{S}}\mid\overleftarrow{\mathrm{S}})}{\Pr(\overrightarrow{\mathrm{S}})}\,. \tag{A.9}$$

This enables me to write:

$$E = \sum_{\{S_i\}}\Pr(\overleftarrow{\mathrm{S}}\,,\,\overrightarrow{\mathrm{S}})\Big[\log(\Pr(\overrightarrow{\mathrm{S}}\mid\overleftarrow{\mathrm{S}})) - \log(\Pr(\overrightarrow{\mathrm{S}}))\Big] \tag{A.10}$$

Using Eqs. (2.3) and (2.4), I may reexpress the second term in Eq. (A.10);

$$-\sum_{\{S_i\}}\Pr(\overleftarrow{\mathrm{S}}\,,\,\overrightarrow{\mathrm{S}})\log(\Pr(\overrightarrow{\mathrm{S}})) = \lim_{L\to\infty}\Big[-\sum_{\{S_i\}}\Pr(\mathrm{S_{-L}},\cdots,\mathrm{S_{-1}},\mathrm{S_0},\mathrm{S_1},\cdots,\mathrm{S_{L-1}})\times \log\,[\,\Pr(\mathrm{S_0},\cdots,\mathrm{S_{L-1}})\,]\,\Big]\,. \tag{A.11}$$

The sum is understood to run over all possible values of all the S_i's. The sum over the "past" S_i's — all S_i with $i < 0$ has no effect since the probabilities are normalized. With this observation, we see that the above equation is nothing more than the entropy rate of an L-cylinder in the limit that L goes to infinity. Thus,

$$\text{Second Term} = \lim_{L\to\infty} H(L) \ . \tag{A.12}$$

Now, the first term in Eq.(A.10) may be written

$$\text{First Term} = \sum_{\{S_i\}} \Pr(\ \overleftarrow{\mathrm{S}}\ ,\ \overrightarrow{\mathrm{S}}\ ) \log(\Pr(\ \overrightarrow{\mathrm{S}}\ \mid\ \overleftarrow{\mathrm{S}}\ )) =$$

$$\lim_{L\to\infty}\Bigg[\sum_{\{S_i\}} \Pr(\mathrm{S}_{-\mathrm{L}},\cdots,\mathrm{S}_{-1},\mathrm{S}_0,\cdots,\mathrm{S}_{\mathrm{L}-1}) \log\left[\ \Pr(\mathrm{S}_0\cdots\mathrm{S}_{\mathrm{L}-1}|\mathrm{S}_{-1},\cdots,\mathrm{S}_{-\mathrm{L}})\ \right]\Bigg] \ . \tag{A.13}$$

Factoring the probability in the argument of the logarithm as we did in Eq. (A.4), this may be written;

$$\text{First Term} = \sum_{\{S_i\}} \Pr(\ \overleftarrow{\mathrm{S}}\ ,\ \overrightarrow{\mathrm{S}}\ ) \log_2(\Pr(\ \overrightarrow{\mathrm{S}}\ \mid\ \overleftarrow{\mathrm{S}}\ )) = \tag{A.14}$$

$$\lim_{L\to\infty}\Bigg[\sum_{\{S_i\}} L\Pr(\mathrm{S}_{-\mathrm{L}},\cdots,\mathrm{S}_{-1},\mathrm{S}_0,\mathrm{S}_1,\cdots,\mathrm{S}_{\mathrm{L}-1}) \times \log_2\left[\ \Pr(\mathrm{S}_{\mathrm{L}-1}|\mathrm{S}_{\mathrm{L}-2}\mathrm{S}_{\mathrm{L}-3}\cdots,\mathrm{S}_{-\mathrm{L}})\ \right]\Bigg] \tag{A.15}$$

$$= \lim_{L\to\infty} [-Lh_\mu.] \ . \tag{A.16}$$

The last equality follows from Eq. (2.7).

So, collecting the first and second terms, I have obtained the desired result:

$$\begin{aligned} E &= \mathrm{MI}(\ \overrightarrow{S}\ ;\ \overleftarrow{S}\ ) \\ &= \lim_{L\to\infty} [H(L) - h_\mu L] \ . \end{aligned} \tag{A.17}$$

So there.

Appendix B

Calculation of h_μ from an ϵ-machine

The goal of this appendix is to derive eqs. (3.12), an expression for the entropy density h_μ in terms of the probability of the causal states and their transitions. We begin with the expression for the entropy density, eq. (2.7):

$$h_\mu = \lim_{L\to\infty} H[S_L|S_{L-1}S_{L-2}\cdots S_1] \; . \tag{B.1}$$

Using the definition of the conditional entropy, eq. (1.20), this may be rewritten as:

$$h_\mu = \lim_{L\to\infty} \sum_{s_L} \sum_{\{s_{L-1}\}} \Pr(s_L, s^{L-1}) \log_2 \Pr(s_L|s^{L-1}) \; , \tag{B.2}$$

where s_L denotes the single spin variable at site L and S_{L-1} denotes the block of $L-1$ spins from sites 1 to $L-1$.

The causal states $\mathcal{S}$ partition the set $\{s^{L-1}\}$ in the sense that each s^{L-1} belongs to one and only one causal state. As a result we may reexpress the sum as follows:

$$h_\mu = \lim_{L\to\infty} \sum_{s_L} \sum_{i} \left(\sum_{s_{L-1}\in\mathcal{S}_i} \Pr(s_L, s^{L-1}) \log_2 \Pr(s_L|s^{L-1}) \right) \; . \tag{B.3}$$

Causal states were defined in eq. (3.5) such that two blocks of spins s_i^{L-1} and s_j^{L-1} belong to the same causal state if and only if $\Pr(S_L|s_i^{L-1}) = \Pr(S_L|s_i^{L-1})$. This observation enables us to perform the sum inside the large parenthesis in eq. (B.3). Each term in the argument of the logarithm

is identical, since all the s_{L-1}'s belong to the same causal state. As a result, we can pull this term outside the sum:

$$h_\mu \;=\; \lim_{L\to\infty} \sum_{s_L} \sum_i \log_2 \Pr(s_L|\mathcal{S}_i) \left(\sum_{s_{L-1}\in\mathcal{S}_i} \Pr(s_L, s^{L-1}) \right) . \tag{B.4}$$

Note that since we are interested in the $L \to \infty$ limit, we need only concern ourselves with recurrent causal states. The summation inside the large parenthesis just has the effect of adding up the probabilities of all the i_{L-1}'s in the $i^{\underline{\text{th}}}$ causal state:

$$\sum_{s_{L-1}\in\mathcal{S}_i} \Pr(s_L, s^{L-1}) \;=\; \Pr(s_L, \mathcal{S}_i) \tag{B.5}$$

Plugging this result into eq. (B.4), we immediately obtain

$$h_\mu \;=\; -\sum_{\{\mathcal{S}_i\}} \sum_{s\in\mathcal{A}} \Pr(s, \mathcal{S}_i) \log_2 \Pr(s|\mathcal{S}_i) \; . \tag{B.6}$$

which is eq. (3.12)