Chapter 1

Introduction

The central problem considered in this thesis is, loosely speaking, that of understanding the behavior of long pattern occurrences in realizations of random processes in discrete time. A typical question we will be asking is the following: *Suppose we observe the outcome of a binary random process; how long does it take until a certain pattern of zeros and ones first appears?* Questions of this type arise naturally in several areas, sometimes because of their theoretical interest and sometimes in applications. Here are four representative examples.

i. *Poincaré recurrence.* Here one asks questions about the *reappearance* of an initial pattern generated by the process. Does it always reappear? When it does, how long does it take? This problem and its ramifications are important in the study of dynamical systems in ergodic theory. In Chapter 3 we will ask what happens when we look for longer and longer such initial patterns – how much longer do we have to wait each time?

ii. *String matching.* Given two finite strings that are generated independently by the same process, what is the length of their longest common (contiguous) substring? This question arises in DNA sequence matching and in string searching algorithms in computer science. As we will see in Chapters 3 and 4, there is a natural "duality" relationship between questions about longest-match lengths, and questions about the first occurrence of random patterns.

iii. *Typicality.* In a long realization of a stationary ergodic process there are "typical" patterns that tend to appear often and "atypical" ones that only appear

rarely. This observation was made by Shannon in his landmark 1948 paper [62]. What is the length and the relative frequency of typical patterns? In Chapter 2 we generalize Shannon's original answers for these questions to real-valued (or more general) processes, and also to the case when distortion is allowed in the patterns.

iv. *Data compression.* Shannon's observation of typical patterns provides a precise way to quantify how much structure there is in a "message" produced by a random "source." How can we take advantage of this structure to do "compression," i.e., to describe long messages efficiently? The celebrated Lempel-Ziv family of data compression algorithms is based on exploiting this structure. In Chapter 5 we extend this idea further to the case of lossy data compression.

This list is by no means exhaustive. Several related questions are mentioned in Section 1.3 below.

As we shall see later, there is a common theme at the heart of all these problems – a strong connection between the geometry along a single realization and the probabilistic structure of the underlying process that produced it, in particular, with the entropy of that process. We can interpret this connection in the "big picture" by saying that it provides yet another snapshot of the sample-path picture of stochastic processes, added to the many other such properties that have come to form a major part of the foundation of modern probability theory over the past 50 years.

1.1 The Question of Recurrence

In order to get a better idea of the flavor of our problems and the ideas involved in solving them, we present here a concrete example of a question that is tackled in detail in Chapter 3. We will try to illustrate three points: (1) the motivation for the problem and the intuition underlying the analysis; (2) the natural way in which the entropy enters when we calculate probabilities of patterns along a realization; (3) the connection between pattern matching and data compression.

1.1.1 Recurrence and Entropy

Suppose we observe a doubly-infinite realization $\boldsymbol{x} = (\ldots, x_{-1}, x_0, x_1, x_2, \ldots)$ produced by a stationary ergodic process $\boldsymbol{X} = \{X_n \ ; \ n \in \mathbb{Z}\}$, which takes values in a finite alphabet A. Write x_i^j for the substring of $\boldsymbol{x}$ between positions i and j

$$x_i^j \stackrel{\triangle}{=} (x_i, x_{i+1}, \ldots, x_j), \quad -\infty \leq i \leq j \leq \infty,$$

and similarly X_i^j for the vector of random variables $(X_i, X_{i+1}, \ldots, X_j)$. For a fixed integer n we consider the pattern $x_1^n = (x_1, \, x_2, \, \ldots, \, x_n)$ formed by the first n symbols produced by $\boldsymbol{X}$, and we ask how far back into the past one has to look before seeing the same pattern appear again. More precisely, we define R_n, the *recurrence time* for x_1^n, as the first position $k \geq 1$ for which $x_{-k+1}^{-k+n} = x_1^n$:

$$R_n \ = \ \inf\{k \geq 1 \ : \ x_{-k+1}^{-k+n} = x_1^n\}.$$

If we increase the length of the pattern we are looking for, then, clearly, the time we have to wait will increase, which implies that for every fixed realization $\boldsymbol{x}$ the recurrence time R_n increases with n. Our main question here is: *How fast does R_n increase?*

To gain some intuition we first try to understand what happens in the simplest case. Suppose $\boldsymbol{X}$ is a sequence of independent and identically distributed (i.i.d.) binary random variables, with each $X_n = 1$ with probability p, or $X_n = 0$ with probability $(1-p)$. Below we show an example of a realization from $\boldsymbol{X}$, with two recurring strings x_1^4 and x_1^5 and corresponding recurrence times $R_4 = 14$ and $R_5 = 26$.

$$\cdots \underbrace{0\,0\,1\,0\,1}_{R_5=26} 1\,0\,1\,1\,1\,0\,1 \overbrace{0\,0\,1\,0}^{R_4=14} \, 0\,0\,1\,1\,0\,1\,0\,1\,1\,0 \overbrace{\underbrace{0\,0\,1\,0\,1}_{x_1^5}}^{x_1^4} \cdots$$

Conditional on the value of x_1, say $x_1 = 1$, the distribution of the recurrence time R_1 is exponential, with mean $1/p$. Thus, R_1 is concentrated around the reciprocal of the probability of the recurring symbol and has exponential tails away from its mean.

What about R_n for general n? Although its distribution is more complicated in this case, it is not hard to show that conditional on the recurring pattern x_1^n, the

mean of R_n is still equal to the reciprocal of the probability of that pattern

$$E(R_n \,|\, X_1^n = x_1^n) = \frac{1}{P(x_1^n)}, \tag{1.1}$$

where P denotes the distribution of $\boldsymbol{X}$. Now what is this probability? If n is large, there will be roughly np ones and $n(1-p)$ zeros in x_1^n, so that $P(x_1^n) \approx p^{np}(1-p)^{n(1-p)}$. Since this decays exponentially with n it suggests that, at least on the average, R_n increases exponentially with n. Moreover, looking at the exponent of decay of $P(x_1^n)$, we see that

$$-\frac{1}{n}\log P(x_1^n) \approx -\frac{1}{n}\log\left(p^{np}(1-p)^{n(1-p)}\right) = H, \tag{1.2}$$

where[1] $H = -p\log p - (1-p)\log(1-p)$ is the *entropy rate* of the process $\boldsymbol{X}$. This, then, suggests that R_n increases exponentially with a rate in the exponent given by the entropy rate of $\boldsymbol{X}$ and, indeed, it is probably not very surprising that the above informal argument can easily be made rigorous to show that

$$\lim_{n\to\infty} \frac{1}{n}\log R_n \;=\; H \quad \text{a.s.} \tag{1.3}$$

What is somewhat remarkable, though, is that each one of the above steps is essentially valid in full generality – for every finite-valued stationary ergodic process: A theorem of Kac from 1947 [34] says that (1.1) remains *verbatim* true for every stationary ergodic $\boldsymbol{X}$. This can be used to conclude (not trivially – see Theorem 3.1 in Chapter 3) that the asymptotic behavior of R_n is the same as that of $1/P(X_1^n)$, in that

$$\lim_{n\to\infty}\left[\frac{1}{n}\log R_n - \frac{1}{n}\log\frac{1}{P(X_1^n)}\right] = \lim_{n\to\infty}\frac{1}{n}\log\left[R_n P(X_1^n)\right] = 0 \quad \text{a.s.}, \tag{1.4}$$

and the Shannon-McMillan-Breiman theorem [13] states that (1.2) also remains true in this case

$$\lim_{n\to\infty} -\frac{1}{n}\log P(X_1^n) = H \quad \text{a.s.} \tag{1.5}$$

[1] Here and throughout this thesis log denotes the logarithm taken to base 2, and $\log_e$ denotes the natural logarithm.

where the entropy rate H of $\boldsymbol{X}$ is now defined by $H \stackrel{\triangle}{=} \lim_n E[-\log P(X_1 \,|\, X_{-n}^0]$. Combining (1.4) and (1.5) we recover (1.3) in complete generality!

1.1.2 Second-Order Results

After seeing that the rate in the exponent of the recurrence times R_n converges, with probability one, to a constant (the entropy rate H), there is a natural sequence of further questions we would like to ask, including:

i. What is the *rate of convergence* to the H in (1.3)?

ii. What is the *asymptotic distribution* of the deviations away from H?

iii. What is the *variance* of these deviations?

The way we will answer these questions in Chapter 3 is by refining the steps we took in the strategy that gave us (1.3). The main intuition we gained there was that, in a strong asymptotic sense, R_n, the recurrence time for the pattern X_1^n is close to the reciprocal of the probability $P(X_1^n)$ of that pattern. First we will show that the formal connection between R_n and $1/P(X_1^n)$ given in (1.4) can be strengthened to

$$\lim_{n\to\infty} \frac{1}{\sqrt{n}} \log[R_n P(X_1^n)] \;=\; 0 \quad \text{a.s.} \tag{1.6}$$

Then, looking at $-\log P(X_1^n)$ a little more carefully and assuming for a moment that $\boldsymbol{X}$ is i.i.d., we see that $-\log P(X_1^n)$ can be rewritten as an ordinary random walk

$$-\log P(X_1^n) = \sum_{i=1}^{n} [-\log P(X_i)], \tag{1.7}$$

so that its asymptotic behavior can be described in detail by the classical limit theorems for partial sums of i.i.d. random variables. For example, combining equations (1.6) and (1.7) with the classical central limit theorem immediately yields

$$\frac{\log R_n - nH}{\sqrt{n}} \xrightarrow{\mathcal{D}} N(0, \sigma^2) \tag{1.8}$$

with $\sigma^2 = \mathrm{Var}(-\log P(X_1))$, answering our questions (ii) and (iii) above [" $\xrightarrow{\mathcal{D}}$ " denotes convergence in distribution]. This can be viewed as a central-limit-theorem-type refinement to the strong-law-of-large-numbers statement of (1.3). Similarly, a simple application of the law of the iterated logarithm gives

$$\limsup_{n\to\infty} \frac{\log R_n - nH}{\sqrt{2n \log_e \log_e n}} = \sigma \quad \text{a.s.}, \tag{1.9}$$

providing the pointwise rate of convergence in (1.3) and answering question (i).

In Chapters 2 and 3 we show that the independence assumption can be significantly relaxed, and the same strategy works for a large class of processes with memory.

1.1.3 Recurrence and Data Compression

How did the question of the asymptotic behavior of R_n first arise?

In 1989, in an attempt to understand the exact compression performance of some variants of the Lempel-Ziv data compression algorithm, Wyner and Ziv [69] discovered the connection between recurrence times and entropy described in (1.3). One of the central ideas in their paper was, instead of considering the actual algorithms directly, to introduce and analyze an idealized coding scenario, a simple version of which we describe below.

Suppose an encoder and a decoder, me and you, say, have been communicating for a long time so that presently we share a very long, in fact *infinitely* long, common database $X_{-\infty}^0 = (\ldots, X_{-1}, X_0)$ produced by some stationary ergodic "source" $\boldsymbol{X}$. My task as the encoder is to describe to you the "message" X_1^n consisting of the next n symbols produced by $\boldsymbol{X}$, and I want to find a way to utilize somehow the "common information" $X_{-\infty}^0$ we share in order to describe X_1^n more efficiently.

My idea is, rather than describing X_1^n to you directly, I will look in the database $X_{-\infty}^0$, find the first position R_n where a copy of the message X_1^n appears, and tell you that position. From this information you can easily recover X_1^n by looking in the database and reading off the string $(X_{-R_n+1}, X_{-R_n+2}, \ldots, X_{-R_n+n})$.

Is this a good idea? Since all I have to tell you is R_n, my description consists of approximately $\log R_n$ bits (in general it takes about $\log k$ bits to describe an integer k), and from this you can recover a message of length n symbols, giving a compression

ratio of approximately

$$\frac{\log R_n}{n} \quad \text{bits per symbol.}$$

As we saw in (1.3) this ratio converges to the entropy rate of $\boldsymbol{X}$, implying that the compression performance of this simple-minded scheme is asymptotically optimal!

Although of no practical use in itself, this result provides the main technical ingredient in proving the optimality of the so-called Sliding-Window Lempel-Ziv algorithm [84][71], probably the most popular compression algorithm in use today. Moreover, Wyner and Ziv's idea of reducing the study of a practical algorithm to that of an idealized coding scenario was a very significant contribution to our intuitive understanding of the workings of several Lempel-Ziv schemes. Since then, this reduction has been exploited by a number of authors and has ultimately lead not only to a better understanding of the existing methods, but also to several new, practical data compression algorithms.

In Section 1.2.2 below we will push this connection a little further; we will discuss extensions of the Lempel-Ziv idea to lossy data compression, and motivate our subsequent results in Chapter 5.

1.2 Three More Questions

Next we outline three more questions that are addressed later in this thesis, and we highlight some of our relevant results from Chapters 2–5.

1.2.1 Waiting Times

Consider the following variation of the recurrence times problem: Instead of asking how long it takes before the first *re*appearance of the initial pattern generated by some random process, we ask how long it takes before the first *approximate appearance* of a random pattern generated independently by a different process.

For the sake of simplicity, consider two i.i.d. binary processes $\boldsymbol{X} = \{X_n \,;\, n \in \mathbb{Z}\}$ and $\boldsymbol{Y} = \{Y_n \,;\, n \in \mathbb{Z}\}$, with distributions P and Q, respectively. We will measure the closeness between finite realizations from $\boldsymbol{X}$ and $\boldsymbol{Y}$ by the proportion of positions

where they agree, so we define the *Hamming distortion* between x_1^n and y_1^n by

$$\rho_n(x_1^n, y_1^n) \;=\; \frac{1}{n}\sum_{i=1}^{n} I\{x_i = y_i\}, \quad x_1^n,\; y_1^n \in \{0,1\}^n,\; n \geq 1, \tag{1.10}$$

where $I\{x_i = y_i\}$ is the indicator function of the event $\{x_i = y_i\}$. For any binary string x_1^n and any distortion level $D \in [0,1]$ we let $B(x_1^n, D)$ denote the distortion-ball of radius D around x_1^n:

$$B(x_1^n, D) = \{y_1^n \in \{0,1\}^n \,:\, \rho_n(x_1^n, y_1^n) \leq D\}.$$

Given two realizations of $\boldsymbol{X}$ and $\boldsymbol{Y}$ and a $D \in [0,1]$, our quantity of interest here is the *waiting time* $W_n(D)$ until a D-close version of x_1^n first appears somewhere in y_1^∞:

$$W_n(D) \;=\; \inf\{k \geq 1 \,:\, y_k^{k+n-1} \in B(x_1^n, D)\}.$$

Intuitively, it seems natural to expect that the asymptotic behavior of $W_n(D)$ as $n \to \infty$ would not be very different from that of R_n, so we ask: *To what extent does $W_n(D)$ behave like R_n?*

In Chapter 4 this question is addressed (and answered), and the analysis follows essentially the same strategy as the one employed to analyze the behavior of R_n:

i. First, we prove that the waiting time $W_n(D)$ until we find a D-close match for X_1^n can be approximated by the reciprocal of the probability $Q(B(X_1^n, D))$ of finding such a match (see Theorem 4.1, Chapter 4):

$$\log W_n(D) \approx -\log Q(B(X_1^n, D)).$$

ii. Then we show that, asymptotically, $-\log Q(B(X_1^n, D))$ behaves as a random walk (Theorems 2.4 and 2.5, Chapter 2), just like $-\log P(X_1^n)$ did in the case of R_n.

Although these two steps closely parallel the corresponding recurrence times results in (1.6) and (1.7), the techniques used to prove them had to be different in this case. One of the difficulties can be spotted easily from the fact that we cannot expand $-\log Q(B(X_1^n, D))$ as random walk like we did with $-\log P(X_1^n)$ in (1.7). In fact,

it is not even clear *a priori* that $-\log Q(B(X_1^n, D))$ will have the same asymptotic behavior as $-\log P(X_1^n)$.

Chapter 2 is devoted to showing that the same behavior does indeed persist, in that the probabilities $Q(B(X_1^n, D))$ decay exponentially and their deviations from the limiting exponent are asymptotically those of a random walk. These results provide natural generalizations of the Shannon-McMillan-Breiman theorem and its refinements to general processes (taking more than a discrete set of values) and to the case when distortion is allowed.

Combining, as before, (i) and (ii) with the classical limit theorems for partial sums of i.i.d. random variables we obtain analogs of (1.3), (1.8) and (1.9): From the strong law of large numbers it follows that the waiting times $W_n(D)$ increase exponentially with probability one,

$$\lim_{n\to\infty} \frac{1}{n} \log W_n(D) \;=\; R(P, Q, D) \quad \text{a.s.}, \tag{1.11}$$

where the rate in the exponent $R(P, Q, D)$ can be explicitly identified as the solution to a variational problem in terms of the entropies of $\boldsymbol{X}$ and $\boldsymbol{Y}$. Similarly, using the central limit theorem and the law of the iterated logarithm we get analogs for (1.8) and (1.9), respectively.

1.2.2 Lossy Data Compression

In many engineering applications where large amounts of data are to be stored or transmitted, compression is an important component. Often, in order to reduce the storage or transmission requirements, we are willing to tolerate a certain amount of error in the reconstructed data – for example, think of a large image database where each image is compressed by a factor of, say, 50:1, and can be recovered not perfectly, but with a small amount of visual distortion. The following question will be addressed in Chapter 5: *Is there an easy way to extend the Lempel-Ziv idea to the case when distortion is allowed, to obtain a practical lossy compression scheme based on pattern matching?*

The great success of the Lempel-Ziv family of algorithms has been mainly due to two reasons. First, they are low complexity algorithms that can be simply implemented (they are, for example, implemented on almost every personal computer in

use today). Since efficient string matching has been very well studied by computer scientists over the past several decades, there are, by now, a number of very efficient algorithms that can be readily used in the context of compression.

The second reason for their practical success is that Lempel-Ziv schemes are *universal* – they assume essentially zero prior knowledge about the distribution of the source to be compressed. The trick they employ to overcome this lack of knowledge comes down to the idea of using the message itself as a codebook. For example, in the idealized coding scenario described in relation to recurrence times (Section 1.1.3 above), we assumed that the encoder and decoder shared an infinitely long database that had the same distribution as the source, and that the next part of the message was described by a pointer into that database.

There is, therefore, an implicit assumption that plays a key role in the success of these compression algorithms, namely, that the optimal (lossless) description of some random message is in terms of a codebook with the same distribution as the message itself. Unfortunately, this assumption is *not true* in the lossy case, and one is forced to consider codebooks generated according to different distributions.

To understand the situation better we follow Wyner and Ziv's example [69] and turn to an idealized coding scenario: Consider an encoder and a decoder sharing a common infinite database $Y_1^\infty = (Y_1, Y_2, \ldots)$, generated by some i.i.d. binary process $\boldsymbol{Y}$ with distribution Q. Suppose that the encoder's task is to communicate a message X_1^n, generated by a different i.i.d. binary process $\boldsymbol{X}$ of distribution P, to the decoder, within some prescribed distortion D (with respect, say, to Hamming distortion $\{\rho_n\}$ as defined in (1.10)). The encoder's strategy is, as before, to look through the database until the first time when a D-close match of X_1^n is found, and then tell the decoder the position $W_n(D)$ of this first match. To describe $W_n(D)$ it takes roughly $\log W_n(D)$ bits, so the compression achieved by this simple code equals

$$\frac{\log W_n(D)}{n} \quad \text{bits per symbol.}$$

As we saw in (1.11), this converges to $R(P, Q, D)$, so different choices of the database distribution yield different limiting compression ratios. The bad news here is that, unlike in the case of lossless compression, $R(P, Q, D)$ is *not* in general minimized by choosing the database to be of the same distribution as the source, i.e., taking $Q = P$. On the other hand, the optimal compression ratio for $\boldsymbol{X}$ with respect to Hamming

distortion at level D (given by the *rate-distortion function* $R(D)$ of $\boldsymbol{X}$) satisfies

$$R(D) = \inf_Q R(P, Q, D)$$

so that the problem is that we do not know *a priori* how to choose the best database distribution in order to minimize $R(P, Q, D)$.

In Chapter 5 we describe a new lossy version of Lempel-Ziv coding that gets around this problem by maintaining not just one, but *multiple* databases at the encoder and the decoder, and chooses which one to use at each stage in a "greedy" manner. The new algorithm is demonstrated to have asymptotically optimal compression performance (Theorem 5.2), and we argue that its complexity and redundancy characteristics are comparable to those of its lossless counterpart.

1.2.3 Match Lengths and DNA Template Matching

In the analysis of DNA or protein sequences the following problem is of interest: Suppose we have a template $(X_1, X_2, \ldots)$ and a long but finite database sequence $Y_1^m = (Y_1, Y_2, \ldots, Y_m)$. *What is the length of the longest initial portion X_1^ℓ of the template that matches within distortion D somewhere in the database?* By a "match" here we mean that there exists a contiguous substring $Y_{j+1}^{j+\ell}$ of the database such that the distortion between X_1^ℓ and $Y_{j+1}^{j+\ell}$ is at most D, with respect to, say, Hamming distortion. Given two realizations of the processes $\boldsymbol{X}$ and $\boldsymbol{Y}$ producing the above template and database, respectively, we write $L_m(D)$ for this maximal match-length:

$$L_m(D) = \sup\{\ell \geq 1 \ : \ y_{j+1}^{j+\ell} \in B(x_1^\ell, D), \ \text{ for some } \ j = 0, 1, \ldots, m-1\}.$$

Intuitively it seems that there is some connection between the match lengths $L_m(D)$ and the waiting times $W_n(D)$. We would expect that the database length m is essentially the same as the waiting time for $(X_1, \ldots, X_{L_m(D)})$, that is, if $n = L_m(D)$ then $W_n(D)$ should be approximately equal to m, and vice versa. Taking this analogy a step further, we might be tempted to replace m by $W_n(D)$ and n by $L_m(D)$ in our asymptotic results about waiting times, and hope that they remain valid.

We will see in detail in Chapters 3 and 4, that this intuition is essentially correct but it is not trivial to justify. For example, replacing m by $W_n(D)$ and n by $L_m(D)$

in (1.11) we obtain (see Theorem 4.2 in Chapter 4)

$$\lim_{m\to\infty} \frac{\log m}{L_m(D)} \;=\; R(P,Q,D) \quad \text{a.s.} \tag{1.12}$$

Similarly, all second-order results about $W_n(D)$ give us corresponding results for $L_m(D)$, providing a complete picture of the asymptotic behavior of $L_m(D)$.

1.3 History

Some general remarks about the history of the results we have been discussing are in order here. More detailed references to specific or more recent results are given at appropriate points in the subsequent chapters.

In ergodic theory, the question of what we called Poincaré recurrence was first raised by Poincaré in 1899 [59]. A very nice exposition of the long history of the results that followed, and also of the connection with the infamous H-theorem of Boltzmann, are presented in Petersen's text [55]. Kac's theorem was proved in 1947 [34]; alternative proofs can be found in [55][69].

Within probability theory, recurrence properties have been very important since at least as far back as the late 1930's. Doeblin and Harris both identified recurrence as the key concept in analyzing the asymptotic behavior of Markov processes; see Meyn and Tweedie's book [48] for a modern exposition. In particular, the idea of approximating the waiting time for an event by the reciprocal of its probability appears already in Doeblin's work on continued fractions in 1940 [24], in Bellman and Harris' (1951) work on the Ehrenfest model [10], and also in Harris' (1952) paper [31] on recurrence in Markov chains. At the cost of more restrictive assumptions, these authors go a step further and essentially show that the distribution of the waiting time for a rare event A is approximately exponential, with mean equal to the probability of A. Recent work in this direction is reported by Galves and Schmitt [27] who also provide an extensive list of references.

Closer to our approach, the use of $-\log P(X_1^n)$ or a similar random walk as an approximating sequence was employed by Ibragimov [32] and by Philipp and Stout [57, Chapter 9] in proving refinements to the Shannon-McMillan-Breiman theorem; by Barron [7] in proving the Shannon source coding theorem in the almost sure sense; and

by Algoet and Cover [2] in an elementary proof of the Shannon-McMillan-Breiman theorem.

The notion of typicality was introduced by Shannon in his famous 1948 paper [62] that founded the field of information theory. Our calculation of the probability of a typical sequence that lead to equation (1.2) was taken, essentially *verbatim*, from the discussion preceding Theorem 3 in [62]. There, Shannon showed that for every stationary ergodic Markov chain $\boldsymbol{X}$ with a finite number of states,

$$-\frac{1}{n}\log P(X_1^n) \to H \quad \text{in probability.} \tag{1.13}$$

McMillan [47] showed that (1.13) holds for every stationary ergodic process, and Breiman [13] strengthened McMillan's result to the almost sure convergence result we saw in (1.5). Meanwhile, first Yushkevich [77] in 1953 and then Ibragimov [32] in his well-known 1962 paper proved a central limit theorem refinement of (1.13). More on the history of further work in this direction is given in Chapter 2.

Turning to applications, the first explicit connection between match lengths and entropy seems to have been made in 1985 by Pittel [58], whose results are phrased in terms of path lengths in random trees. Aldous and Shields [1] pointed out the relationship between randomly growing trees and data compression, and Szpankowski [66] made explicit the equivalence between match lengths along random sequences and feasible paths in random trees.

Recurrence times in relation to data compression first appeared in Willems' work [67] and also in Wyner and Ziv's 1989 paper [69], where they (implicitly) introduced the idealized coding scenario we saw in Section 1.1.3. Wyner and Ziv [69] discovered (1.3) and the corresponding result for waiting times (without distortion), and these were formally established by Ornstein and Weiss [53] and by Shields [63], respectively, using methods from ergodic theory. Extensive references to subsequent work of refining and generalizing these results are given in Chapters 3 and 4.

In connection with DNA sequence analysis, results about asymptotics of match lengths arising from string matching problems can found in the work of Karlin and Ost [35], Pevzner, Borodovsky and Mironov [56], Arratia and Waterman [5], and Dembo, Karlin and Zeitouni [20]. Some of these results can be viewed as natural generalizations of the classical Erdös-Rényi laws of large numbers, as discussed by Arratia, Gordon and Waterman in [4]. Finally we mention that related questions

about string searching algorithms in computer science have been studied by Guibas and Odlyzko [29] and Jacquet and Szpankowski [33], among many others.

1.4 About This Thesis

1.4.1 Theory and Applications

Our initial motivation for this work was to gain a better understanding of the workings of the Lempel-Ziv family of data compression algorithms. Our introduction to the problem was through Wyner and Ziv's 1989 paper [69]; there, they isolated two very interesting theoretical questions (the questions about the asymptotic behavior of recurrence and waiting times), and demonstrated that the performance of the practical algorithms can be determined from the answers to these questions. Subsequently, researchers in several communities outside information theory found these problems also to be of theoretical interest and expanded on Wyner and Ziv's work. In the process of generalizing the original results to the case when distortion is allowed, further theoretical questions arose which led to the generalizations of the Shannon-McMillan-Breiman theorem and its refinements that we present in Chapter 2. These results, in turn, provided the intuition that was missing in order to solve an important practical problem, that of finding a practical extension of the Lempel-Ziv idea to the case of lossy compression – see Chapter 5.

In summary, a real practical application gave rise to some interesting theoretical questions, whose solutions may have significant impact in practice.

1.4.2 Organization

The rest of the thesis is organized as follows.

In Chapter 2 we describe the Shannon-McMillan-Breiman theorem, its refinements (by Yushkevich [77], Ibragimov [32], and Philipp and Stout [57]), and their generalizations to the case when distortion is allowed (by Łuczak and Szpankowski [45], Yang and Kieffer [75], and Dembo and Kontoyiannis [21]).

In Chapter 3 we address the problem of recurrence times in stationary processes, and we show the asymptotic behavior of the recurrence times R_n can be deduced from that of the random walk $-\log P(X_1^n)$. This, combined with the results presented in

Chapter 2, gives us a complete asymptotic description of R_n. Corresponding results are proved for certain longest match-lengths M_m along a realization, by exploiting a nice duality relationship between R_n and M_m.

Chapter 4 contains analogous results about waiting times, both with and without distortion. We first show that the behavior of the waiting times $W_n(D)$ can be deduced from that of the Q-probabilities of distortion balls $B(X_1^n, D)$, and then we apply our results from Chapter 2 to read-off the asymptotics of $W_n(D)$. Again, corresponding results are proved for the match lengths $L_m(D)$ via duality.

In Chapter 5 we address the problem of finding an extension of the Lempel-Ziv data compression algorithm that has asymptotically optimal compression performance, and is also implementable in practice. We introduce a new lossy variant of Lempel-Ziv, we prove its asymptotic optimality, and we argue that its complexity and redundancy characteristics are comparable to those of its lossless counterpart.

The contributions of this thesis are briefly summarized in Chapter 6, where we also mention some promising future research directions.

Finally in Appendix A we give the proofs of some of the more technical results from Chapters 2–5.

1.5 Notation

Here we state some notation and definitions that will remain in effect throughout this thesis. Although most of these are repeated (at least once) somewhere else, we also collect them here for easy reference.

- $\boldsymbol{X} = \{X_n \ ; \ n \in \mathbb{Z}\}$ denotes a stationary process with values in some space $(A, \mathcal{A})$, and distribution determined by the measure P on the product space $(A^\infty, \mathcal{A}^\infty)$.
- Similarly, $\boldsymbol{Y} = \{Y_n \ ; \ n \in \mathbb{Z}\}$ denotes a stationary process with values in some space $(\hat{A}, \hat{\mathcal{A}})$, and distribution determined by the measure Q on $(\hat{A}^\infty, \hat{\mathcal{A}}^\infty)$.

- For integers $-\infty \le i \le j \le \infty$, we denote by X_i^j the vector of random variables $(X_i, X_{i+1}, \ldots, X_j)$. Similarly, for a sequence $(x_n)_{n\in\mathbb{Z}}$ of elements from a set A, x_i^j denotes the part of the sequence between positions i and j.

- $\boldsymbol{x}$ denotes an infinite realization $\boldsymbol{x} = x_{-\infty}^{\infty} \in A^{\infty}$ of the process $\boldsymbol{X}$; similarly, $\boldsymbol{y}$ denotes a realization $\boldsymbol{y} = y_{-\infty}^{\infty} \in \hat{A}^{\infty}$ of $\boldsymbol{Y}$.

- "log" denotes the logarithm taken to base 2, and "$\log_e$" denotes the natural logarithm.

- $H(X) \stackrel{\triangle}{=} -\sum_x P(x) \log P(x)$ denotes the entropy (in bits) of the discrete random variable X, distributed according to the probability mass function P.

- $H(P)$ denotes the entropy rate (in bits) of the process $\boldsymbol{X}$ with distribution P, and is defined by
$$H(P) \;=\; \lim_{n\to\infty} \frac{1}{n}\, H(X_1^n).$$
If $\boldsymbol{X}$ is stationary then, equivalently, $H(P) \;=\; \lim_n E[-\log P(X_0 \,|\, X_{-n}^{-1})]$.

- $H(P\|Q)$ denotes the relative entropy (in bits) between the two probability measures P and Q, and is defined by
$$H(P\|Q) = \begin{cases} \int dP \log \frac{dP}{dQ}, & \text{when } \frac{dP}{dQ} \text{ exists} \\ \infty, & \text{otherwise.} \end{cases}$$

- $I(X;Y) \stackrel{\triangle}{=} H(P_{(X,Y)} \| P_X \times P_Y)$ denotes the mutual information (in bits) between the random variables X and Y, where P_X and P_Y denote the marginals of X and Y, respectively, and $P_{(X,Y)}$ is their joint distribution.

- ρ is some fixed measurable function $\rho : A \times \hat{A} \to [0, \infty)$, and $\{\rho_n\}$ is a sequence of single-letter distortion measures $\rho_n : A^n \times \hat{A}^n \to [0, \infty)$ defined by
$$\rho_n(x_1^n, y_1^n) = \frac{1}{n} \sum_{i=1}^{n} \rho(x_i, y_i), \qquad x_1^n \in A^n,\; y_1^n \in \hat{A}^n,\; n \ge 1.$$

- $R(D)$ is the rate-distortion function (in bits) of the process $\boldsymbol{X}$, with respect to the sequence of distortion measures $\{\rho_n\}$ and at distortion level D; it is defined

by

$$R(D) = \lim_{n\to\infty} \frac{1}{n} \inf_{\pi_n \in \mathcal{Q}_n} I(X_1^n; Y_1^n)$$

where $\mathcal{Q}_n$ is the space of all joint distributions π_n for (X_1^n, Y_1^n), such that $\int \rho_n(x_1^n, y_1^n) d\pi_n(x_1^n, y_1^n) \leq D$ and the X_1^n-marginal of π_n is the same as the original distribution of X_1^n.

- $H_e(X)$, $H_e(P)$, $H_e(P\|Q)$, $I_e(X;Y)$ and $R_e(D)$ denote the entropy, entropy rate, relative entropy, mutual information and rate-distortion function in *nats* rather than in bits, i.e., they have the same definitions as the corresponding functionals without the subscript e, but with the logarithms to base 2 replaced with natural logarithms.