Three lectures on information theory

Michael Hochman

These are expanded (but still rough) notes of lectures delivered for advanced undergraduate students at the Hebrew University in September 2011. The notes primarily cover source coding from a theoretical point of view: Shannon's bound and Kraft's inequality, basic properties of entropy, entropy of stationary processes, Shannon-McMillan ("AEP") for mixing Markov chains,, and the Lempel-Ziv algorithm. Some background material on probability and Markov chains is included as well. Please send comments to mhochman {ampersand} math.huji.ac.il.

Contents

1 Introduction

Information theory was essentially founded by Claude Shannon in his 1948 paper "A mathematical theory of communication". Although technology have changed a great deal since then the basic model of how communication is carried out has not: a sender wants to communicate a message to a receiver through a noisy channel. The process can be decomposed into the following schematic diagram:

$$\text{Sender: } \omega \xrightarrow[\text{coding}]{\text{source}} \omega' \xrightarrow[\text{coding}]{\text{channel}} \omega'' \xrightarrow[\text{channel}]{\text{noisy}} \omega''' \xrightarrow[\text{and decoding}]{\text{error correction}} \text{Receiver}$$

Source coding "compresses" the message to be sent, making the message that will be transmitted as short as possible; channel coding introduces redundancy, so that noise can be detected or corrected; the channel introduces (random) noise into the message; and the receiver tries to reverse these processes.

There are many variations, e.g. the receiver may be able to request that data be resent. There are also related theories which can fit into the diagram such as encryption.

We will focus here on the first stage – source coding. Besides its role in communication across a channel, it has obvious applications to data storage, and many applications in other fields (for example it is closely related to entropy theory of dynamical system).

There are two variants, of source coding:

> **Lossless**: we want to recover the message exactly, e.g. when compressing digital data. Example: the "zip" compression program.
>
> **Lossy**: we allow an ε-fraction of error when decoding. For example in music or voice compression we are willing to lose certain frequencies which are inaudible to us (hopefully). Example: mp3

In these lectures we focus on compression of the lossless kind.

First, after a short discussion of different types of codes, we will identify theoretical limits on the amount of compression of a random source, and show that the optimal average compression can be achieved within one bit. We then turn to codings of sequences of source symbols assuming that they are drawn from a stationary process (e.g. i.i.d. process or Markov chain) and show that the asymptotic per-symbol compression can be achieved with arbitrary small penalty. These coding methods are explicit but are not efficient, and most importantly they depend on knowledge of the statistics of the source. In the last part we will discuss universal coding, and present the Lempel-Ziv compression algorithm which gives optimal asymptotic coding for any stationary input. Along the way we will derive the definition of entropy and discuss its formal properties.

2 Codes

Notation and conventions Let Σ^n be the set of length-n sequences with symbols in Σ, for $\sigma \in \Sigma^n$ write $\sigma = \sigma_1\sigma_2 \ldots \sigma_n$ and abbreviate $\sigma = \sigma_1^n$. Let $\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n$, the set of finite sequences, including the empty word $\emptyset$. Let $|a|$ denote the length of $|a|$. If $a, b \in \Omega^*$ we write ab for their concatenation. All logarithms are to base 2.

Let Ω be a finite set of "messages". We often assume $\Omega = \{1, \ldots, n\}$.

Definition 2.1. A **code** is a map $c : \Omega \to \{0,1\}^*$.

The word "code" also sometimes refers to the range of c. Sometimes it is useful to analyze codes $c : \Omega \to \Sigma$ for other alphabets Σ but $\Sigma = \{0,1\}$ is the most common case and we focus on it; the general case is similar.

Worst case bound: if we use words of length n to code messages, i.e. $c : \Omega \to \{0,1\}^\ell$ for some ℓ, and assuming c is 1-1, then $|\Omega_0| \leq 2^\ell$, or: $\ell \geq \log|\Omega|$ (remember $\log = \log 2$).

Probabilistic model: $0 \leq p(\omega) \leq 1$ is a probability distribution on Ω, i.e. $\sum p(\omega) = 1$.

Definition 2.2. The **mean coding length** of c is $\sum_{\omega\in\Omega} p(\omega)|c(\omega)|$.

This is the expected coding length, and if we imagine repeatedly coding random (independent) messages this is also the long-term number of bits per message which we sill use (this observation uses the law of large numbers).

Definition 2.3. Given a code $c : \Omega \to \{0,1\}^*$ its extension to Ω^* is $c^* : \Omega^* \to \{0,1\}^*$ given by $c^*(\omega_1, \ldots, \omega_n) = c(\omega_1) \ldots c(\omega_n)$. We usually drop the superscript and just write $c(\omega_1 \ldots \omega_n)$.

Definition 2.4. c is uniquely decodable if c^* is 1-1.

Non-example: Let $\Omega = \{u, v, w\}$. and the code $u \mapsto 0$ $v \mapsto 00$ $w \mapsto 1$. since $c(uu) = c(v)$.

Definition 2.5. A marker $u \in \{0,1\}^*$ is a word such that if u appears a a subword of $a \in \{0,1\}^*$ then its occurrences do not overlap.

Example: Any single symbol is a marker; 10 is a marker; 1010 is not a marker because it has overlapping occurrences in 101010.

Definition 2.6. A **Marker code** is a code $c(a) = uc'(a)$, where u is a marker, $c'(a)$ does not contain u, and c' is injective (but $(c')^*$ does not have to be).

Example: A code with codewords $01, 011, 0111$ (the marker is 0); or $10000, 10001, 10010, 10011$ (the marker is 100).

Marker codes are uniquely decodable because given an output $a = c(\omega_1^n)$, we can uniquely identify all occurrences of the marker, and the words between them can be decoded with $(c')^{-1}$ to recover a.

Marker codes also have the advantage that if we receive only a portion of the coding of the $a = c^*(\omega_1\omega_2 \ldots \omega_N)$ (for example if the first k symbols are deleted and we receive $b = a_{k+1} \ldots a_{|c^*(\omega_1^N)|}$), we can still recover part of the message – the same decoding procedure works, except possibly the first few symbols of b will not be decoded.

Although they are convenient, marker codes are harder to analyze, and are suboptimal when coding single messages (although when coding repeated messages they are asymptotically no worse than other codes). Instead we will turn to another class of codes, the prefix codes.

A word a is a prefix of a word b is $b = aa'$ for some (possibly empty) word a'. Equivalently $a = b_1 \ldots b_{|a|}$.

Definition 2.7. A **Prefix code** is a code c such that if $\omega \neq \omega'$ then $c(\omega)$ is not a prefix of $c(\omega')$ or vice versa.

Example: $0, 10, 11$ but in this example, you need to read from the beginning

Example: $01, 00, 100, 101, 1100, 1101, 1110, 1111$

Lemma 2.8. *Prefix codes are uniquely decodable.*

Proof. Suppose $u = u_1 \ldots u_n \in \{0,1\}^n$ is written as $u = c(\omega_1) \ldots c(\omega_m) = c(\omega'_1) \ldots c(\omega'_{m'})$. Let $r \geq 0$ be the largest integer such $\omega_i = \omega'_i$ for $1 \leq i < r$. We want to show that $r = m = m'$. Writing $k = \sum_{i<r} |c(\omega_i)|$ this is equivalent to showing that $k = n$. If $k \neq n$ then $u_{k+1} \ldots u_n = c(\omega_r) \ldots c(\omega_m) = c(\omega'_r) \ldots c(\omega'_{m'})$ is a non-empty word (in particular $r \leq m$ and $r \leq m$) and both $c(\omega_r)$ and $c(\omega'_r)$ are prefixes of $u_{k+1} \ldots u_n$. This implies that one is a prefix of the other. Since $\omega_r \neq \omega'_r$ by definition of r, this contradicts the fact that c is a prefix code. Hence $k = n$ as required. □

Our goal is to construct codes with optimal average coding length. Since average coding length is determined by the lengths of codewords, not the codewords themselves, what we really need to know is which lengths $\ell_1, \ldots, \ell_n$ admit a uniquely decodable codes with these lengths.

Theorem 2.9. *(Generalized Kraft inequality) Let $\ell_1, \ldots, \ell_n \geq 1$. Then the following are equivalent:*

1. $\sum 2^{-\ell_i} \leq 1$
2. There is a prefix code with lengths ℓ_i.
3. There is a uniquely decodable code with lengths ℓ_i.

Proof. 2⇒3 was the previous lemma.

1⇒2: Let $L = \max \ell_i$ and order $\ell_1 \leq \ell_2 \leq \ldots \leq \ell_n = L$.

It is useful to identify $\bigcup_{i \leq L} \{0,1\}^i$ with the full binary tree of height L, so each vertex has two children, one connected to the vertex by an edge marked 0 and the other by an edge marked 1. Each vertex is identified with the labels from the root to the vertex; the root corresponds to the empty word and the leaves (at distance L from the root) correspond to words of length L.

We define codewords $c(i) = a^i$ by induction. Assume we have defined $a^i, i < k$ with $|a^i| = \ell_i$. Let

$$A_i = \{a^i b \,:\, b \in \{0,1\}^{L-\ell_i}\}$$

Thus A_i is the set of leaves descended from a^i, or the set of words of length L of which a^i is a prefix. We have $|A_i| = 2^{L-\ell_i}$. The total number of leaves descended form $a^1, \ldots, a^{k-1}$ is

$$|\bigcup_{i<k} A_i| \leq \sum_{i<k} |A_i| = \sum_{i<k} 2^{L-\ell_i} < 2^L$$

The strict inequality is because $\sum 2^{-\ell_i} \leq 1$, and the sum above includes at least one term less than the full sum.

Let $a \in \Sigma^L \setminus \bigcup A_i$ and $a^k = a_1 \dots a_{\ell_k}$ the length-ℓ_k prefix of a. For $i < k$, a^i if a^i is a prefix of a^k then, since $\ell_i \leq \ell_k$, a^k is a child of a^i and so $a \in A_i$, a contradiction. If a^k is a prefix of a^i then since $\ell_i \leq \ell_k$ we have $a^i = a^k$ and the same arrive at the same contradiction. Therefore $a^1, \dots, a^k$ is a prefix code.

3⇒1: Suppose c is uniquely decodable. Fix m. Then

$$(\sum 2^{-\ell_i})^m = \sum_{(i_1 \dots i_m) \in \Omega^m} 2^{-\sum_{j=1}^m \ell_{i_j}} = \sum_{(i_1,\dots,i_m) \in \Omega^m} 2^{-|c(i_1,\dots,i_m)|}$$

divide the codewords according to length:

$$= \sum_{\ell=1}^{Lm} \sum_{\omega \in \Omega^{\leq m} \,:\, c(\omega)=\ell} 2^{-\ell} \leq \sum_{\ell=1}^{Lm} 2^{-\ell} 2^{\ell} = Lm$$

taking m-th roots and $m \to \infty$, this gives (1) . □

3 Optimal coding of random sources

Objective: given (Ω, p), find code c such that $\sum_{\omega \in \Omega} p(\omega) \ell(c(\omega))$ is minimal.

Equivalently: find $\{\ell_\omega\}_{\omega \in \Omega}$ which minimize $\sum p_\omega \ell_\omega$ subject to $\sum 2^{-\ell_\omega} \leq 1$.

Replace the integer variable ℓ_i by continuous real x_i. We have the analogous optimization problem (we can add dummy variables and assume we want $\sum 2^{-x_i} = 1$).

Lagrange equation:

$$L(x, \lambda) = \sum p_\omega x_\omega - \lambda \left(\sum 2^{-x_\omega} - 1 \right)$$

so

$$\begin{aligned} p_\omega - \lambda 2^{-x_\omega} \log 2 &= 0 \\ \sum 2^{-x_\omega} &= 1 \end{aligned}$$

Summing the first equation over ω, and using $\sum p_\omega = 1$, we find

$$\lambda = 1/\log 2$$

so

$$x_\omega = -\log p_\omega$$

and the "expected coding length" at the critical point is

$$-\sum p_\omega \log p_\omega$$

Note: we still don't know this is the "global" max.

Definition 3.1. Entropy $H(p_1, \ldots, p_n) = -\sum p_i \log p_i$.

Concavity: convex and strictly convex functions f, sufficient condition in $\mathbb{R}$: $f'' \leq 0$ (< 0), uniqueness of maxima of strictly concave function on an interval.

Example: log is concave

Theorem 3.2. *If c is uniquely decodable then the expected coding length is $\geq H(p)$ and equality is achieved if and only if $p_i = 2^{-\ell_i}$.*

Proof. Let ℓ_i be the coding length of i. We know that $\sum 2^{-\ell_\omega} \leq 1$. Consider

$$\begin{aligned} \Delta &= H(p) - \sum p_\omega \ell_\omega \\ &= -\sum p_\omega \left(\log p_\omega + \ell_\omega\right) \end{aligned}$$

Let $r_\omega = 2^{-\ell_\omega} / \sum 2^{-\ell_\omega}$, so $\sum r_\omega = 1$ and $\ell_\omega \geq -\log r_\omega$ (because $\sum 2^{-\ell_\omega} \leq 1$).

$$\begin{aligned} \Delta &\geq -\sum p_\omega \left(\log p_\omega - \log r_\omega\right) \\ &= -\sum p_\omega \left(\log \frac{p_\omega}{r_\omega}\right) \\ &= \sum p_\omega \left(\log \frac{r_\omega}{p_\omega}\right) \end{aligned}$$

Using the concavity of the logarithm,

$$\geq \log \sum p_\omega (\frac{r_\omega}{p_\omega}) = \log 1 = 0$$

Equality occurs unless $r_\omega / p_\omega = 1$. □

Theorem 3.3 (Achieving optimal coding length (almost))**.** *There is a prefix code whose average coding length is $H(p_1, \ldots, p_n) + 1$*

Set $\ell_\omega = \lceil -\log p_\omega \rceil$. Then

$$\sum 2^{-\ell_\omega} \leq \sum 2^{-\log p_\omega} = \sum p_\omega = 1$$

and since $\ell_\omega \leq -\log p_\omega + 1$, the expected coding length is

$$\sum p_\omega \ell_\omega \leq H(p) + 1$$

Thus **up to one extra bit we achieved the optimal coding length**.

4 Review: Probability spaces and random variables

This is a quick review of probability for those who never have taken a course on it.

A discrete probability space is a finite set Ω whose points are "world states".

For example if we draw two cards from a deck then Ω could consist of all pairs of distinct cards: $\{(i, j) : 1 \leq i, j \leq 52\,,\, i \neq j\}$.

An event is a subset of Ω. For example, the event

$$A = \{\text{first card is jack of spades}\}$$

is a set of the form $\{i\} \times \{1, \ldots, 52\} \subseteq \Omega$, where i represents jack of spades.

A **probability distribution** on Ω is a probability vector $p = (p_\omega)_{\omega \in \Omega}$, that is $p_\omega \geq 0$ and $\sum_{\omega \in \Omega} p_\omega = 1$. We also write $p(\omega) = p_\omega$. We shall often use the word distribution to refer to a probability vector in general.

The probability of an event is

$$P(A) = \sum_{\omega \in A} p(\omega)$$

For example if all point have equal mass then $p(\omega) = 1/|\Omega|$ and $p(A) = |A|/|\Omega|$.

Example: In the card example above we give all pairs equal weight, then $p(i, j) = (52 \cdot 51)^{-1}$.

Properties:

$$p(\emptyset) = 0 \qquad p(\Omega) = 1$$

$$0 \leq p(A) \leq 1$$

$$A \subseteq B \quad \Longrightarrow \quad p(A) \leq p(B)$$

$$p(A \cup B) = p(A) + p(B) - p(A \cap B)$$

$$p(A \cup B) \leq p(A) + p(B)$$

$$A \cap B = \emptyset \quad \Longrightarrow \quad p(A \cup B) = p(A) + p(B)$$

$$p(\Omega \setminus A) = 1 - p(A)$$

and more generally for finite sums:

$$A_i \cap A_j = \emptyset \quad \Longrightarrow \quad p(\cup A_i) = \sum p(A_i)$$

A **random variable** is a function $X : \Omega \to \Omega'$. Often but not always $\Omega' = \mathbb{R}$. We can think of X as a measurement of the world; $X(\omega)$ is the value of the measurement when the world is in state ω.

Example: in the card example, $X(i, j) = i$ would be the value of the first card, and $Y(i, j) = j$ the value of the second card.

We write

$$p(X = x) = p(\{\omega \in \Omega : X(\omega) = x\}) = p(X^{-1}(x))$$

for the probability that X takes the value x. We can also define events using RVs:

$$\{0 \leq X \leq 1\} = \{\omega \in \Omega : 0 \leq X(\omega) \leq 1\}$$

The probability vector $dist(X) = \{p(X = x)\}_x$ is often called the **distribution** of X and can be thought of as a distribution on the range of X. All questions involving only X are determined by this probability vector. We also will abbreviate

$$p(x) = p(X = x) \qquad p(y) = p(Y = y) \qquad p(x, y) = p((X, Y) = (x, y))$$

etc., when it is typographically clear what RV we mean

Given pair X, Y of random variables, we can consider the random variable (X, Y). The **joint distribution** of X and Y is the distribution

$$dist(X, Y) = (p(X = x, Y = y))_{x,y}$$

This may be thought of as a probability distribution on the range of the RV (X, Y). This definition to finite families of random variables.

The **conditional distribution** $dist(X|Y = y)$ of X given $Y = y$ is defined by

$$p(x|y) = p(X = x|Y = y) = \frac{p((X, Y) = (x, y))}{P(Y = y)}$$

or equivalently:

$$P(X = x, Y = y) = p(Y = y) \cdot p(X = x|Y = y)$$

We can think of this as a probability distribution on the set $\{Y = y\}$, determined by restricting p to this set, and considering the random variable $X|_{\{Y=y\}}$.

Example: in the card example suppose all pairs are equally likely. Then $p(X = i) = 1/52$ but $P(X = i|Y = j) = (51 \cdot 52^{-1}/52^{-1} = 1/51$ because on the set $\{Y = j\}$ the value of X is equally likely to be any of the remaining $i \neq j$ but it cannot be $= j$. This corresponds to the intuition that if we choose i, j from a deck one at a time, the second choice is constrained by the first.

Example: independent coin toss and roll of die

Example: coin toss followed by fair or unfair die, depending on head/tail.

two random variables are **independent** if $p(x, y) = p(x)p(y)$. In general a sequence $X_1, \ldots, X_N$ of RVs is independent if for any two disjoint subsets of indices $I, J \subseteq \{1, \ldots, N\}$, we have

$$p((x_i)_{i\in I}, (x_j)_{j\in J}) = p((x_i)_{i\in I}) \cdot p((x_j)_{j\in J})$$

5 Properties of entropy

We already encountered the entropy function

$$H(t_1, \dots, t_k) = -\sum t_i \log t_i$$

We use the convention $0 \log 0 = 0$, which makes $t \log t$ continuous on $[0,1]$.

The entropy of a finite-valued random variable X is

$$H(X) = h(dist(X)) = -\sum_x p(X = x) \log p(X = x)$$

Note: $h(X)$ depends only on the distribution of X so if X' is another RV with the same distribution, $H(X) = H(X')$.

Interpretation: $H(X)$ is the amount of uncertainty in X, or the amount of randomness; it is also how hard it is to represent the value X on average.

Lemma 5.1. *$H(\cdot)$ is strictly concave.*

Remark if p, q are distributions on Ω then $tp + (1-t)q$ is also a distribution on Ω for all $0 \le t \le 1$.

Proof. Let $f(t) = -t \log t$. Then

$$\begin{aligned} f'(t) &= -\log t - 1 \\ f''(t) &= -\frac{1}{t} \end{aligned}$$

so f is strictly concave on $(0, \infty)$. Now

$$H(tp+(1-t)q) = \sum_i f(tp_i+(1-t)q_i) \ge \sum_i (tf(p_i) + (1-t)f(q_i)) = tH(p)+(1-t)H(q)$$

with equality if and only if $p_i = q_i$ for all i. □

Lemma 5.2. *If X takes on k with positive probability. Then $0 \le H(X) \le \log k$. the left is equality if and only if $k = 1$ and the right is equality if and only if $dist(X)$ is uniform, i.e. $p(X = x) = 1/k$ for each of the values.*

Proof. The inequality $H(X) \ge 0$ is trivial. For the second, note that since H is concave on the convex set of probability vectors it has a unique maximum. By symmetry this must be $(1/k, \dots, 1/k)$. □

Definition 5.3. The **joint entropy** of random variables X, Y is $H(X,Y) = H(Z)$ where $Z = (X, Y)$.

The **conditional entropy** of X given that another random variable Y (defined on the same probability space as X) takes on the value y is is an entropy associated to the conditional distribution of X given $Y = y$, i.e.

$$H(X|Y=y) = H(dist(X|Y=y) = -\sum_x p(X=x|Y=y)\log p(X=x|Y=y)$$

The **conditional entropy** of X given Y is the average of these over y,

$$H(X|Y) \quad = \quad \sum_y p(Y=y)\cdot H(dist(X|Y=y))$$

Example: X, Y independent $\frac{1}{2}, \frac{1}{2}$ coin tosses.

Example: X, Y correlated coin tosses.

Lemma 5.4. .

1. *$H(X,Y) = H(X) + H(Y|X)$*
2. *$H(X,Y) \geq H(X)$ equality iff Y func. of X.*
3. *$H(X|Y) \leq H(X)$ equality iff X, Y indep.*

Proof. Write

$$\begin{aligned}
H(X,Y) &= -\sum_{x,y} p((X,Y)=(x,y))\log p((X,Y)=(x,y)) \\
&= \sum_y p(Y=y)\sum_x p(X=x|Y=y)\left(-\log\frac{p((X,Y)=(x,y))}{p(Y=y)} - \log p(Y=y)\right) \\
&= -\sum_y p(Y=y)\log p(Y=y)\sum_x p(X=x|Y=y) \\
&\quad -\sum_y p(Y=y)\sum_x p(X=x|Y=y)\log p(X=x|Y=y) \\
&= H(Y) + H(X|Y)
\end{aligned}$$

Since $H(Y|X) \geq 0$, the second inequality follows, and it is an equality if and only if $H(X|Y=y) = 0$ for all y which Y attains with positive probability. But this occurs if and only if on each event $\{Y = y\}$, the variable X takes one value. This means that X is determined by Y.

The last inequality follows from concavity:

$$\begin{aligned}
H(X|Y) &= \sum_y p(Y=y)H(dist(X|Y=y)) \\
&\leq H(\sum_y p(Y=y)dist(X|Y=y)) \\
&= H(X)
\end{aligned}$$

and equality if and only if and only if $dist(X|Y=y)$ are all equal to each other and to $dist(X)$, which is the same as independence. □

There is a beautiful axiomatic description of entropy as the only continuous functional of random variables satisfying the conditional entropy formula. It is formulated in the following exercise:

Exercise. Suppose that $H_m(t_1,\ldots,t_m)$ are functions on the space of m-dimensional probability vectors, satisfying

1. $H_2(\cdot,\cdot)$ is continuous,
2. $H_2(\frac{1}{2},\frac{1}{2})=1$,
3. $H_{k+m}(p_1,p_2,\ldots p_k,q_1,\ldots,q_m)=(\sum p_i)H_k(p'_1,\ldots,p'_k)+(\sum q_i)H_m(q'_1,\ldots,q'_k)$, where $p'_i=p_i/\sum p_i$ and $q'_i=q_i/\sum q_i$.

Then $H_m(t)=-\sum t_i\log t_i$.

6 Stationary processes

Recall that we write $X_i^j=X_iX_{i+1}\ldots X_j$, etc.

A **random process** is a sequence of random variables $X_1,\ldots,X_N$, taking values in the same space Ω_0. Given such a sequence it is always possible to assume that the probability space is Ω_0^N with the distribution $dist(X_1^N)$. Conversely, if p is a distribution on Ω^N, define random variables X_n by $X_n(\omega_1^N)=\omega_n$.

We will want to talk about infinite sequences X_1^∞ of random variable. To do so requires defining probability distributions on the space $\Omega_0^{\mathbb{N}}$ of infinite sequences and this requires more sophisticated tools (measure theory). There is a way to avoid this however if we are only interested in questions about probabilities of finite (but arbitrarily long) subsequences, e.g. about the probability of the event $\{X_2^7=X_{12}^{17}\}$. Formally, we define a stochastic process X_1^∞ to be a sequence of probability distributions p^N on Ω_0^N, $N\in\mathbb{N}$, with the property that if $M>N$ then for any $x_1^N\in\Pi_0^N$,

$$p^N(x_1^N)=\sum_{x_{N+1}^M\in\Omega_0^{M-N}}p^M(x_1^M)$$

Given any finite set of indices $I\subseteq\mathbb{N}$ and $x_i\in\Omega_0$, $i\in I$, let $N>\max I$, and define

$$p(X_i=x_i\text{ for }i\in I)=\sum_{y_1^N\in\Omega_0^N\,:\,y_i=x_i\text{ for }i\in I}p^N(y_1^N)$$

Using the consistency properties of p^N it is easy to see that this is independent of N. This definition allows us to use all the usual machinery of probability on the sequence X_1^∞ whenever the events depend on finitely many coordinates $1\ldots N$; in this case we are working in the finite probability space Ω_0^N but we can always enlarge N as needed if we need to consider finitely many additional coordinates.

We are mainly interested in the following class of processes:

Definition 6.1. A **stationary process** is a process X_1^∞ such that for every $a_1 \dots a_k$, and every n,

$$p(X_1^k = a_1^k) = p(X_n^{n+k-1} = a_1^k)$$

That is, if you know that the sequence a occurred, this does not give you any information about **when** it occurred.

Example: If we flip a fair coin over and over and we are told that the sequence 1111 occurred, we can't tell when this happened.

Example: If we flip a fair coin for the first hundred flips and then flip an unfair one (biased towards 1) from then on and are told that 1111 occurred, it is more likely to come from times ≥ 100 than times < 100. This process is not stationary.

Example (product measures): More generally let $\Omega = \{1, \dots, n\}$ and $q = (q_i)_{i \in \Omega}$ a probability vector and let

$$q(X_i^j = x_i^j) = \prod_{k=i}^{j} q_{x_k}$$

This defined a stationary stochastic process called **the i.i.d. process with marginal** q.

Remark "Physically" stationary processes correspond to physical processes in equilibrium. Although many real-world phenomena are not in equilibria generally given enough time they approach an equilibrium and stationary processes are a good idealization of the limiting equilibrium, and useful for its analysis.

Our next goal is to discuss the entropy of stationary processes.

Lemma 6.2. *If* $a_{n+m} \leq a_n + a_m$ *and* $a_n \geq 0$ *then* $\lim \frac{1}{n} a_n$ *exists and equals* $\inf \frac{1}{n} a_n \geq 0$.

The proof is an exercise!

Lemma 6.3. *For a stationary process,* $\lim_{n\to\infty} \frac{1}{n} H(X_1, \dots, X_n)$ *exists and is equal to* $\lim_{n\to\infty} H(X_1 | X_2^n)$.

Proof. Notice that

$$\begin{aligned} H(X_1^{m+n}) &= H(X_1^m) + H(X_{m+1}^n | X_1^n) \\ &\leq H(X_1^m) + H(X_{m+1}^n) \\ &= H(X_1^m) + H(X_1^{n-m}) \end{aligned}$$

by stationarity. Thus $a_n = H(X_1^n)$ is subadditive and the limit exists.

Now,

$$H(X_1^m) = \sum_{i=1}^{m} H(X_i | X_{i+1}^m) = \sum_{i=1}^{m} H(X_1 | X_2^i)$$

again by stationarity. Also, the sequence $H(X_1|X_2^i)$ is decreasing in i and bounded below, hence $H(X_1|X_2^n) \to h$ for some $h \geq 0$. Therefore we have

$$\lim \frac{1}{n} H(X_1^n) = \lim \frac{1}{n} \sum_{i=1}^{n} H(X_1|X_2^i) = h$$

because the Cesaro averages of a convergent sequence converge to the same limit. □

Definition 6.4. The **mean entropy** $h(X_1^\infty)$ of a stationary process X_1^∞ is the limit above.

Example: if X_i are i.i.d. with marginal p then $H(X_1^m) = mH(p)$; this is because $H(Y,Z) = H(Y) + H(Z)$ when Y, Z are independent, and we can iterate this for the independent sequence $X_1 \ldots X_m$. Therefore $h(X_1^\infty) = H(p)$.

Definition 6.5. Let $c : \Omega_0^N \to \{0,1\}^*$ be a uniquely decodable code. The **per-symbol coding length** of c on a process X_1^N is

$$\frac{1}{N} \sum_{x_1^N \in \Omega^N} p(x_1^N) |c(x_1^N)|$$

i.e. the average coding length divided by N.

If X_1^∞ is stationary then $\frac{1}{N} H(X_1^N) \geq h(X_1^\infty)$, because by subaditivity, h is the decreasing limit of $\frac{1}{N} H(X_1^N)$. Therefore h is a lower bound on the per-symbol coding length.

Corollary 6.6. *If X_1^∞ is a stationary process then for every $\varepsilon > 0$ there it is possible to code X_1^k with per-symbol coding length $h + \varepsilon$.*

Proof. For large enough N we have $H(X_1^N) \leq N(h + \varepsilon/2)$. We have seen that there is a (prefix) code for X_1^N with average coding length $\leq N(h + \varepsilon) + 1$, so the per symbol coding length is $h + \varepsilon/2 + 1/N < h + \varepsilon$ when N is large enough. □

Thus for stationary processes it is possible to asymptotically approach the optimal coding rate.

There are several drawbacks still. First, we reached within ε of the per-symbol optimal length but we chose ε in advance, once the code is fixed it will not improve further. Second, the coding is still inefficient in the sense that we need to construct a code for inputs Ω_1^N and this is a very large set; if we were to store the code e.g. in a translation table it would become unreasonably large very quickly. Third, we use knowledge of the distribution of X_1^N in constructing our codes; in many cases we do not have this information available, i.e. we want to code input from an unknown process. All these problems can be addressed: we will see later that there are universal algorithms with asymptotically optimal compression for arbitrary stationary inputs.

7 Markov processes

We will now restrict the discussion to the class of irreducible Markov chains. This is a much smaller class than general stationary processes but it is still quite large and is general enough to model most real-world phenomena.

Definition 7.1. A Markov **transition law** or **Markov kernel** on $\Omega = \{1, \ldots, n\}$ is a function $i \to p_i(\cdot)$ where p_i is a probability vector on Ω.

A transition kernel can also be represented as a matrix

$$A = (p_{ij}) \qquad p_{ij} = p_i(j)$$

A third way to think of this is as weights on the directed graph with vertex set Ω; from vertex i the edges are labeled $p_i(j)$.

Using the third point of view, imagine starting at a (possibly random) vertex i_0 and doing a random walk for N steps (or ∞ steps) according to these edge probabilities. This gives a stochastic process X_1^N such that, given that we are i_n at time n we move to j at time $n+1$ with probability $p_{i_n}(j)$. Assuming our initial distribution was q, the probability of taking the path x_1^k is

$$p(X_1^k = x_1^k) = q(x_1)p_{x_1}(x_2)\ldots p_{x_{k-1}}(x_k)$$

Note that if we define p this way it satisfies (by definition)

$$p(X_k = x_k | X_1^{k-1} = x_1^{k-1}) = \frac{q(x_1)p_{x_1}(x_2)\ldots p_{x_{k-1}}(x_k)}{q(x_1)p_{x_1}(x_2)\ldots p_{x_{k-1}}(x_{k-1})} = p_{x_{k-1}}(x_k)$$

Note that the conditional probability above does not depend on x_1^{k-2}, only on x_{k-1}. Thus **only the present state influences the future distribution; there is no influence from the past.**

Notice that starting from q,

$$\begin{aligned} p(X_2 = x_2) &= \sum_{x_!} p(X_2 = x_x | X_1 = x_1) \cdot p(X_{k-1} = x_{k-1}) \\ &= \sum_{x_1} p_{x_1 x_2} q_{x_1} \\ &= (qA)_{x_2} \end{aligned}$$

where q is a row vector and A as above. More generally

$$\begin{aligned} p(X_3 = x_3) &= \sum_{x_2} p(X_3 = x_3 | X_2 = x_2) p(X_2 = x_2) \\ &= \sum_{x_2} p_{x_2}(x_3) \cdot (qA)_{x_2} \\ &= \sum_{i} (qA)_i p_{i x_3} \\ &= (qA^2)_{x_3} \end{aligned}$$

and in general

$$p(X_k = j) = (qA^k)_j$$

so

$$dist(X_k) = qA^k$$

Definition 7.2. A stationary distribution q is a distribution such that $qA = A$.

Lemma 7.3. *If q is a stationary distribution then the Markov process started from X_1^∞ is stationary.*

Proof. Note that

$$dist(X_k) = qA^k = dist(X_1)$$

Hence

$$p(X_k^m = x_k^m) = p(X_k = x_k) \sum_{i=k+1}^{m} p_{x_{i-1}}(x_i) = p(X_1 = x_k) \sum_{i=k+1}^{m} p_{x_{i-1}}(x_i) = p(X_1^{m-k-1)} = x_1^k$$

□

Lemma 7.4. *If q is stationary then $h(X_1^\infty) = \sum q_i H(p_i(\cdot)) = H(X_2|X_1)$.*

Proof. .

$$\begin{aligned} H(X_1^k) &= H(X_1) + \sum_{i=1}^{k} H(X_i|X_1^{i-1}) \\ &= H(X_1) + \sum_{i=1}^{k} H(X_i|X_{i-1}) \end{aligned}$$

by the Markov property. Since $H(X_i|X_{i-1}) = H(X_2|X_1) = \sum q_i H(p_i(\cdot))$, the result follows by dividing by k and sending $k \to \infty$. □

Assumption $p_{ij} > 0$ for all i, j. We then say that the Markov chain is **mixing**.

This assumption is restrictive; it would suffice to assume irreducibility, i.e. that for every i, j there is some k such that $(A^k)_{i,j} > 0$ (in terms of the random walk on Ω_0 this means that for all i, j there is positive probability to go from i to j in a finite number of steps). All the statements below hold in the irreducible case but the proofs are simpler assuming $p_{i,j} > 0$ for all i, j.

Theorem 7.5. *There is a unique stationary measure q. Furthermore, for any initial distribution q', if Y_1^∞ is the chain started from q', then $q'_n = dist(X'_n) \to q$ exponentially fast in ℓ^1.*

First proof. . From the Perron-Frobenius theorem on positive matrices we know that A has a maximal eigenvalue λ which is simple and is the only eigenvalue with positive eigenvectors. When A acts on the left there is an eigenvector $(1, \dots, 1)$ with eigenvalue 1 so we must have $\lambda = 1$. Thus there is a positive left eigenvector $q = qA$ which we may assume satisfies $\sum q_i = 1$.

Notice that if r is a probability vector then rA is as well, so $rA^k \not\to 0$. We can write $r = sq + tv$, where v is in the sum of generalized eigenspaces of eigenvalues of modulus < 1. Since $vA^k \to 0$, but $rA^k \not\to 0$, we have $s \neq 0$. So

$$rA^k = sq + tvA^k \to sq$$

Since rA^k is a probability vector, $s = 1$, and so $rA^k \to q$. Clearly the convergence is uniform in the space of probability measures (in fact it is exponential). □

Second proof. Let Δ denote the space of probability vectors in $\mathbb{R}^n$. Then $q \mapsto qA$ maps $\Delta \to \Delta$. Note that Δ is closed, so it is a complete metric space.

Consider the subspace $V = \{v : \sum v_i = 0\}$. If $v \in V$ let $I = \{i : v_i > 0\}$ and $J = \{j : v_j < 0\}$. Thus since $v \in V$,

$$\sum_{i \in I} v_i = -\sum_{j \in J} v_j$$

Therefore,

$$\sum_{i \in I} v_i = -\sum_{j \in J} v_j \geq \|v\|_1 > \|v\|_\infty$$

Now, for each k,

$$(vA)_k = \sum_{i \in I} v_i p_{ik} + \sum_{j \in J} v_j p_{jk}$$

each of these sums is $\leq \|v\|_\infty$ in absolute values, their signs are opposite, and each is $\geq \delta \|v\|_\infty$, where

$$\delta = \min p_{ij} > 0$$

Therefore,

$$(vA)_k \leq (1 - \delta) \|v\|_\infty$$

Now if $q, q' \in \Delta$ then $q - q' \in V$ so

$$\|qA - q'A\|_\infty = \|(q - q')A\|_\infty \leq (1 - \delta) \|q - q'\|_\infty$$

so $A : \Delta \to \Delta$ is a contraction. Hence there is a unique fixed point q, satisfying $q = qA$, and for any $q' \in A$, $q'A^k \to q$ exponentially fast. □

Theorem 7.6 (Law of large numbers)**.** *Let $f = f(x_0 \dots x_k)$ be a function $f : \Omega^k \to \mathbb{R}$. Let $Y_i = f(X_i \dots X_{i+k})$ and let $a = \sum p(Y = y) \cdot y$ be the average of Y. Then*

$$p\left(\left|\frac{1}{M}\sum_{m=1}^{M} Y_m - a\right| > \varepsilon\right) \to 0 \qquad \textit{as } M \to \infty$$

Proof. Replacing f by $f - a$ we can assume $f = 0$.

Lemma 7.7 (Chevyshev inequality)**.** *If Z is a real random variable with mean 0 and $\sum p(Z = z) \cdot z^2 = V$ then $p(|Z| \geq \delta) < \frac{V}{\delta^2}$.*

Proof. $p(|Z| \geq \delta) = p(Z^2 \geq \delta^2)$. If we denote this value by π then clearly

$$\begin{aligned} V &= \sum_z p(Z = z) \cdot z^2 \\ &\geq \sum_{z:z^2 \geq \delta^2} p(Z = z) \cdot z^2 \\ &\geq \sum_{z:z^2 \geq \delta^2} p(Z = z)\delta^2 \\ &= p(Z^2 \geq \delta^2)\delta^2 \\ &= p(|Z| \geq \delta)\delta^2 \qquad \square \end{aligned}$$

Returning to the proof of the theorem, we want to show that

$$p\left(\left(\frac{1}{M}\sum_{m=1}^{M} Y_m\right)^2 > \varepsilon^2\right) \to 0 \qquad \text{as } M \to \infty$$

Note

$$\left(\frac{1}{M}\sum_{m=1}^{M} Y_m\right)^2 = \frac{1}{M^2}\sum_{r,s=1}^{M} Y_r Y_s$$

Lemma 7.8. $\mathbb{E}Y_r Y_s \to 0$ *as* $s - r \to \infty$*.*

Proof. We know that $dist(X_s^{s+k}|X_r = x_r) \to dist(X_s^{s+k})$ as $s - r \to \infty$, and so the same is true for $dist(Y_s|X_r = x_r)$. Since

$$\begin{aligned} \mathbb{E}Y_r Y_s &= \sum_{y_r, y_s} p((Y_r, Y_s) = (y_r, y_s)) \cdot y_r \cdot y_s \\ &= \sum_{y_r, y_s} p(Y_r = y_r)p(Y_s = y_s|Y_r = y_r) \cdot y_r \cdot y_s \\ &= \sum_{y_r} p(Y_r = y_r) \cdot y_r \sum_{y_s} p(Y_s = y_s|Y_r = y_r) \cdot y_s \\ &= \sum_{x_r^{r+s}} p(Y_r = f(x_r^{r+s})) \cdot f(x_r^{r+s}) \cdot \sum_{y_s} p(Y_s = y_s|X_r^{r+k} = x_r^{r+k}) \cdot y_s \end{aligned}$$

Each of the inner sums converges to $p(Y_s = y_s)$ as $s - r \to \infty$. So the inner sums converge to $\mathbb{E}Y_s = 0$. Thus the whole sum converges to 0. □

Returning to the proof of the theorem, let $\delta > 0$ and let $d \in \mathbb{N}$ such that $\mathbb{E}(Y_r Y_s)| < \delta$ if $|s - r| > d$. Now,

$$\begin{aligned} \mathbb{E}\left(\frac{1}{M}\sum_{m=1}^{M} Y_m\right)^2 &= \mathbb{E}\frac{1}{M^2}\sum_{r,s=1}^{M} Y_r Y_s \\ &= \frac{1}{M^2}\sum_{r,s=1}^{M} \mathbb{E}Y_r Y_s \end{aligned}$$

as $M \to \infty$, all but dM of the pairs $1 \leq r, s \leq M$ are such that $|r - s| \leq d$, so

$$\begin{aligned} &\leq \frac{dM \left\|f\right\|_{\infty}^{2}}{M^2} + \sum_{1\leq r,s\leq M\,,\,|r-s|>d} |\mathbb{E}(Y_r Y_s)| \\ &\leq \frac{d \left\|f\right\|_{\infty}^{2}}{M^2} + \frac{\delta M^2}{M^2} \end{aligned}$$

This is $< 2\delta$ when M is large enough. Now apply the first lemma. □

8 The Shannon-McMillan Theorem

The LLN says that most sequences for a stationary process have the same “statistics” (e.g. each symbol or block of symbols appear in roughly the right proportion). The following theorem says that in fact most sequences of a given length have **roughly the same probability**, and this probability is determined by the mean entropy of the process.

We assume now that X_1^∞ is a stationary mixing Markov chain, and $h = h(X_1^\infty)$ is its mean entropy.

Theorem 8.1 (Shannon-McMillan)**.** *For every $\varepsilon > 0$, if N is large enough, then there is a set $T_N \subseteq \Omega^N$ (“typical” points) such that $p(T_N) > 1 - \varepsilon$ and for every $x_1^N \in T_N$,*

$$2^{-(h-\varepsilon)N} \leq p(x_1^N) \leq 2^{-(h+\varepsilon)N}$$

In particular, for large enough N,

$$2^{(h-\varepsilon)N} \leq |T_N| \leq 2^{(h+\varepsilon)N}$$

and for every pair of words in T_N have probabilities within a factor of $4^{\varepsilon N}$ of each other.

Proof for independent processes. Let us first discuss the first claim in the case $p = q^{\mathbb{N}}$ is a product measure with marginal $q = (q_\omega)_{\omega\in\Omega_0}$. Then for any x_1^N,

$$p(x_1^N) = \prod_{i=1}^{N} q_{x_i} = \prod_{\omega\in\Omega_o} q_\omega^{\#\{1\leq i\leq N\,:\,x_i=\omega\}}$$

Taking logarithms and dividing by N,

$$-\log p(x_1^N) = -\sum_{\omega\in\Omega_0}\frac{1}{N}\#\{1\leq i\leq N\,:\,x_i=\omega\}\log q_\omega$$

By the LLN, with high probability,

$$|\frac{1}{N}\#\{1\leq i\leq N\,:\,x_i=\omega\}-q_\omega|<\varepsilon$$

This proved the claim. □

Proof for Markov chains.

$$\begin{aligned}\frac{1}{N}\log p(x_1\dots x_N) &= \log\big(q(x_1)p_{x_1}(x_2)p_{x_2}(x_3)\cdot\ldots\cdot p_{x_{N-1}}(x_N)\big)\\ &= \frac{1}{N}\log q(x_1)+\frac{1}{N}\sum_{i=1}^{N-1}\log p_{x_i}(x_{i+1})\end{aligned}$$

By the LLN, on a set T_N of probability $p(T_N)>1-\varepsilon$ the average above is within ε of the mean value, which is

$$\begin{aligned}\mathbb{E}(\log p_{X_1}(X_2)) &= \sum_{x_1,x_2}p(x_1x_2)\log p_{x_1}(x_2)\\ &= \sum_{x_1}q(x_1)\sum_{x_2}p_{x_1}(x_2)\log p_{x_1}(x_2)\\ &= -\sum q_iH(p_i(\cdot))\\ &= -h(X_1^\infty)\end{aligned}$$

We found that for $x_1^N\in T_N$,

$$-h-\varepsilon\leq\frac{1}{N}\log p(x_1^N)\leq -h+\varepsilon$$

which is the same as

$$2^{-(h-\varepsilon)N}\leq p(x_1^N)\leq 2^{-(h+\varepsilon)N}$$

For the bounds on the size of T_N, note that

$$1\geq p(T_N)\geq|T_N|\cdot 2^{-(h+\varepsilon)N}$$

which gives $|T_N|\leq 2^{(h+\varepsilon)N}$; and

$$1-\varepsilon\leq p(T_N)\leq\sum_{x_1^N\in T_N}p(X_1^N)\leq|T_N|2^{-(h-\varepsilon)N}$$

so for N large, $|T_N|\geq(1-\varepsilon)2^{(h-\varepsilon)N}\geq 2^{(h-2\varepsilon)N}$. Replacing ε by $\varepsilon/2$ gives the claim. □

Remark. The conclusion of the SM theorem is also sometimes called the **asymptotic equipartition property** (AEP) because it says that for large n the seqeuences in Ω_0^n mostly have almost equal probabilities.

Corollary 8.2. *If* $A_N \subseteq \Omega_0^N$ *and* $|A_N| \leq 2^{(h-\varepsilon)N}$ *then* $p(A_N) \to 0$.

Proof. Let $0 < \delta < \varepsilon$ and T_N as above for δ. Then

$$\begin{aligned} p(A_N) &= p(A_N \cap T_N) + p(A_N \setminus T_N) \\ &\leq p(A_N \cap T_N) + p(\Omega_0^N \setminus T_N) \end{aligned}$$

Now, for large enough N we know that $p(\Omega \setminus T_n) = 1 - p(T_n) \leq \delta$. On the other hand $p(x_1^N) \leq 2^{-(h-\delta)N}$ for $x_1^N \in T_N$ so

$$p(A_N \cap T_N) \leq |A_N| 2^{-(h-\delta)N} = 2^{-(\varepsilon-\delta)N} \to 0$$

QED. □

Application to coding. Fix ε and a large N and let T_N be as above. Suppose we code x_1^N by

$$c(x_1^N) = \begin{cases} 0i & x_1^N \in T_N \\ 1j & \text{otherwise} \end{cases}$$

where i is the index of x_1^N in an enumeration of T_N and j is the index in Ω_0^N. We code i with the same number of bits $\log |T_N|$ even if it we can do so with fewer bits (e.g. $i = 1$ is coded as $00\ldots0001$), and similarly j is coded using $\log |\Omega_0|$ bits. Then c is a prefix code since once the first bit is read, we know how many bits remain in the codeword. Since i is represented with $\log |T_N| \leq (h+\varepsilon)N$ bits, and j in $N \log |\Omega_0|$ bits, the expected coding length is

$$\begin{aligned} p(T_N)(1 + (h+\varepsilon)N) + (1 - p(T_N))N \log |\Omega_0| &= (1-\varepsilon)(h + \varepsilon + \frac{1}{N})N + \varepsilon \log \Omega_0 N \\ &\leq (h + \varepsilon + \frac{1 + \log |\Omega_0|}{N})N \end{aligned}$$

so the rate per symbol is $\approx h + \varepsilon$, approaching optimal.

This procedure explains "why" it is possible to code long stationary sequences efficiently: with N fixed and large we are essentially dealing with a uniform distribution on a small set T_N of sequences. This procedure however still uses knowledge of the distribution to construct the set T_N (also the procedure is very inefficient). We next present the algorithm of Lempel and Ziv which is asymptotically optimal on any stochastic process and efficient.

9 The Lempel-Ziv algorithm for universal coding

It is convenient to formulate the following algorithm with an infinite input sequence x_1^∞ and infinite output y_1^∞. The algorithm reads blocks of symbols and outputs blocks. If the input is finite it is possible that the last block of input symbols was too short to produce a corresponding output block. This "leftover" input can be dealt with in many ways, but this is not related to the main goal which is to understand the asymptotics when the input is long. So we ignore this matter.

The following algorithm reads the input and produces (internally) a sequence of words $z^1, z^2, \ldots$ which are a **unique parsing** of the input: that is, for each r,

$$x_1 \ldots x_{n(r)} = z^1 \ldots z^r$$

where $n(r) = \sum_{i=1}^r |z^i|$, and $z^i \neq z^j$ for $i \neq j$. The parsing is constructed as follows. Define $z^0 = \emptyset$ (the empty word). Assuming we have read the input $x_1^{n(r-1)}$ and defined the parsing $z^0, z^1, \ldots, z^{r-1}$ of it, the algorithm continues to read symbols $x_{r(n-1)+1} \ldots x_{r(n-1)+2}, \ldots$ from the input until it has read a symbol $x_{n(r)}$ such that the $x_{n(r-1)+1}^{n(r)}$ does not appear among $z^0, \ldots, z^{r-1}$. It then defines $z^r = x_{n(r-1)+1}^{n(r)}$. By definition this is a unique parsing. Also note that there is a unique index $i_r < r$ such that $z^r = z^{i_r} x_{n(r)}$ (if not then the we can remove the last symbol from z^r and obtain a word which does not appear among $z^0, \ldots, z^{r-1}$ contradicting the definition of z^r).

After defining z^r the algorithm outputs the pair $(i_r, x_{n(r)})$; thus the output is a sequence of pairs as above. We shall discuss below how this pair is encoded in binary bits.

First, here is the algorithm in pseudo-code:

Algorithm 9.1 (Lempel-Ziv coding)**.** *Input:* $x_1 x_2 x_3 \ldots \in \Omega_0^{\mathbb{N}}$*, Output: a sequence* $(i_r, y_r) \in \mathbb{N} \times \{0, 1\}$*,* $r = 1, 2, \ldots$*, with* $i_r < r$*.*

```
    $z^0 := \emptyset$ (the empty word).
  - $n := 0$.
  - For $r := 1, 2, \ldots$ do
      - Let $k$ be the largest integer such that $x_{n+1}^{n+k} = z^m$ for some $m \leq r$.
      - Output $(m, x_{n+k+1})$.
      - $z^{r+1} := x_{n+1}^{n+k+1}$.
      - $r := r + 1$.
      - $n := n + k + 1$.
```

Example Suppose the input is

$$0010100110101111101010$$

The parsing $z^1, z^2, \dots, z^8$ is:

$$0; 01; 010; 011; 0101; 1; 11; 01010$$

and the output is

$$(0,0), (1,1), (2,0), (2,1), (3,1), (0,1), (6,1), (5,0)$$

Notice that the sequence z^r constructed in the course of the algorithm satisfies $x_1 x_2 \dots = z^1 z^2 \dots$ and each (i_r, y_r) output at stage r of the algorithm satisfies $i_r < r$ and $z^r = z^{i_r} y_r$. Thus we can decode in the following way:

Algorithm 9.2 (Lempel-Ziv decoding)**.** *Input* $(i_r, y_r) \in \mathbb{N} \times \{0,1\}$*,* $r = 1, 2 \dots$*, such that* $i_r < r$*; Output* $x_1 x_2 \dots \in \{0,1\}^{\mathbb{N}}$*.*

```
- Let $z^0 := \emptyset$.
- For $r := 1, 2, \dots$ do
    - Let $z^r = z^{i_r} y_r$.
    - Output $z^r$
```

Exercise: Verify that this works in the example above.

Coding the pairs into binary. We want to produce a binary sequence, so we want to code the outputs (i_r, y_r) as binary strings. One way is to write down the binary expansion of the integer i_r followed by the bit y_r but this does not form a prefix code. To fix this we must choose a prefix code for encoding integers. One can either do this abstractly using a version of the Kraft theorem for infinite sets, or use the following methods:

Prefix code for integers: given $i \in \mathbb{N}$ let $a_1 \dots a_k \in \{0,1\}^k$ denote its binary representation, so $k = \lceil \log i \rceil$, and let $b = b_1 \dots b_m$ be the binary representation of k, so $m = \lceil \log k \rceil$. Define

$$\widetilde{c}(i) = b_1 b_1 b_2 b_2 \dots b_m b_m 01 a_1 \dots a_k$$

Decoding: In order to decode $\widetilde{c}(i)$, read pairs of digits (b_i, b_i') until we read the pair $(b_n, b_n') = (0,1)$. Let $b = b_1 \dots b_n$. Let k be the integer with binary representation b. Read k more symbols $a = a_! \dots a_k$ and let i be the word with binary representation a. Output i.

Thus this is a prefix code whose length on input i is

$$|\widetilde{c}(i)| = 2(\lceil \log \lceil \log i \rceil \rceil + 1) + \lceil \log i \rceil = (1 + o(1)) \log k$$

bits.

Also for $u \in \Omega_0$ define $\widetilde{c}(y) = j$, where $\Omega_0 = \{\omega_1, \dots, \omega_{|\Omega_0|}\}$ is some enumeration and $y = \omega_j$. Thus $\widetilde{c}(y) = \lceil \log |\Omega_0| \rceil$.

Analysis From now on assume that the pairs (i_r, y_r) output by the Lempel-Ziv algorithm are encoded as $\widetilde{c}(i_r)\widetilde{c}(y_r) \in \{0,1\}^*$. Since $i_r < r$ this is now a binary string of length $(1 + o(1))\log r$. Let x_1^n be the input to LZ and suppose the algorithm ran r steps, defining the words $z^1, \ldots, z^r$ and outputting r pairs (d_i, y_i), $i = 1 \ldots r$ (encoded as above). We have the estimate

$$c(x_1^n) \le r \cdot (1 + o(1)) \log r$$

The crucial point is to analyze r

Theorem 9.3 (Asymptotic optimality of LZ coding)**.** *Let X_1^∞ be a stationary mixing Markov chain. For every $\varepsilon > 0$,*

$$p(x_1^n \,:\, r \log r > (h + \varepsilon)n) \to 0 \qquad \text{as } n \to \infty$$

Heuristic Suppose the words z^i in the coding are "typical", so there are 2^{hk} words of length k. Then

$$n = \sum_{i=1}^{r} |z^i| = \sum_{\ell} \sum_{i:|Z^i| = \ell} \ell = \sum_{\ell=1}^{L} \ell 2^{h\ell}$$

where L is chosen so that equality holds. Thus $n \approx L2^{hL}$, so $L \le \log n/h$ and

$$r = \sum_{\ell=1}^{L} 2^{h\ell} \approx 2^{hL} \approx n/L \le hn/\log n$$

Hence

$$r \log r \le r \log n \le hn$$

The formal proof will take some time.

Write $c(x_1^n) = z^1 \ldots z^r$ for the partition obtained by the LZ algorithm, so $r = r(x_1^n)$ is random and also depends on n. Fix $\varepsilon > 0$.

For $(i, m) \in \Omega_0 \times \mathbb{N}$, let $r_{i,m}$ denote the number of blocks among $z^1, \ldots, z^r$ which are of length m and are preceded in x_1^n by the state i. Thus

$$r = \sum_{i \in \Omega_0} \sum_{m=1}^{n} r_{i,m}$$

The key to the proof is the following remarkable combinatorial fact. It is convenient to assume that the chain began at time 0, so we can condition on the value of X_0.

Proposition 9.4 (Ziv's inequality)**.** $\log p(x_1^n | x_0) \le -\sum r_{i,m} \log r_{i,m}$.

Note: The right hand side is a purely combinatorial quantity, independent of the transition probabilities of the Markov chain.

Proof. Let j_i denote the index preceding the first symbol of z^i.

$$p(x_1^n|x_0) = \prod_{i=1}^{r} p(z^i|x_{j_i})$$

so

$$\begin{aligned}
\log p(x_1^n|x_0) &= \sum_{i=1}^{r} \log p(z^i|x_{j_i}) \\
&= \sum_{x\in\Omega_0} \sum_{m} \sum_{i:|z^i|=m, x_{j_i}=x} \log p(z^i|x) \\
&= \sum_{x} \sum_{m} r_{x,m} \sum_{i:\ldots} \frac{1}{r_{x,m}} \log p(z^i|x) \\
&\leq \sum_{x} \sum_{m} r_{x,m} \log\left(\frac{1}{r_{x,m}} \sum p(z^i|x) \right)
\end{aligned}$$

by concavity, since in the inner sum there are $r_{x,m}$ terms. But since this is a unique parsing, the inner sum now sums to ≤ 1. So

$$\leq \sum_{x} \sum_{m} r_{x,m} \log(1/r_{x,m})$$

□

Corollary 9.5. $p\left(\sum_x \sum_m r_{x,m} \log r_{x,m} \leq (h+\varepsilon)n\right) \to 1 \qquad$ *as* $n \to \infty$

Proof. By Shannon-McMillan it follows that

$$p(\log p(x_1^n|x_0) < -(h+\varepsilon)n) \to 0 \qquad \text{as } n \to \infty$$

(using the fact that $p(x_1^n) = \sum_{x_0} p(x_0)p(x_1^n|x_0)$; we leave it as an exercise). The corollary is now trivial from the previous proposition. □

Recall that we would like to prove that

$$p\left(r \log r \leq (h+\varepsilon)n\right) \to 1 \qquad \text{as } n \to \infty$$

This follows from the corollary above and the following proposition, which is purely combinatorial:

Proposition 9.6. *Either* $r < (n \log n)^2$ *or else* $r \log r \leq (1+o(1)) \sum_{x,m} r_{x,m} \log r_{x,m}$

Indeed, assuming this is true, in the first situation we have

$$\frac{1}{n} c(x_1^n) = \frac{r \log r}{n} \leq \frac{1}{\log n} \to 0$$

and this implies when n is large that we are coding x_1^n using an arbitrarily small number of bits per symbol. In the alternative case the previous lemma applies and we get the optimality theorem.

We note that the inequality $r\log r \le (1+o(1))\sum_x\sum_m r_{x,m}\log r_{x,m}$ in the proposition does not hold in general. For example one can imagine a unique parsing of x_1^n in which there is one word z^i of each length, $r \sim \sqrt{n}$ and $r\log r \ge \sqrt{n}$, while $\log r_{x,m} = 0$ so $\sum_{x,m} r_{x,m}\log r_{x,m} = 0$. The assumption $r \ge n/(\log n)^2$ precludes such behavior (from the proof one sees that even a much weaker assumption is enough).

The proof of the proposition is requires on two lemmas.

Lemma 9.7. $r \le (1+o(1))\frac{n}{\log n}$.

Proof. Fix r and let $\widehat{n}$ denote the smallest possible value with which this r is consistent. Clearly if ℓ is the length of some z^i then this implies that every word in $\bigcup_{\ell'<\ell}\Omega_0^{\ell'}$ appears among the z^j, since otherwise we could replace z^i with one of these words and obtain a smaller total length . Thus is a k and $m < |\Omega_0|^{k+1}$ such that

$$r = \sum_{i\le k}|\Omega_0|^i + m$$

(note that $k\to\infty$ as $r\to\infty$) and also

$$\widehat{n} = \sum_{i\le k} i|\Omega_0|^i + (k+1)m$$

so

$$r \le \sum_{i\le k}|\Omega_0|^i + m \le \frac{\widehat{n}}{k}$$

On the other hand

$$\widehat{n} \le C(k+1)|\Omega_0|^{k+1}$$

so

$$\log\widehat{n} \le (k+1) + \log(C\cdot(k+1)) = (1+o(1))(k+1) \qquad as\ r\to\infty$$

so

$$r \le (1+o(1))\frac{\widehat{n}}{\log\widehat{n}} \le (1+o(1))\frac{n}{\log n}$$

since $t/\log t$ is increasing for $t\ge 1$. □

Lemma 9.8. *Most of the contribution to r comes from blocks with relatively short length, in the sense that*

$$\sum_{i\in\Omega_0}\sum_{m>n^{\varepsilon/2}} r_{i,m} \le n^{1-\varepsilon/2}$$

Proof. We have

$$\begin{aligned} n &= \sum_i \sum_m m r_{i,m} \\ &\geq \sum_i \sum_{m > n^{\varepsilon/2}} m^{\varepsilon/2} r_{i,m} \\ &= m^{\varepsilon/2} \sum_i \sum_{m > n^{\varepsilon/2}} r_{i,m} \square \end{aligned}$$

Proof of the proposition. We want to show that most of the contribution to the sum $\sum_{i,m} r_{i,m} \log r_{i,m}$ comes from terms with $\log r_{i,m} \sim \log r$. Let

$$I = \{(i,m) \in \Omega_0 \times \mathbb{N} \,:\, \log r_{i,m} > (1-\varepsilon) \log r\}$$

We estimate $\sum_{(i,m) \in I} r_{i,m}$. Let's look at the complementary sum:

$$\begin{aligned} \sum_{(i,m) \notin I} r_{im} \log r_{i,m} &= \sum_i \left(\sum_{m > n^{\varepsilon/2}, \log r_{i,m} \leq (1-\varepsilon) \log r} r_{i,m} + \sum_{m < n^{\varepsilon/2}, \log r_{i,m} \leq (1-\varepsilon) \log r} r_{i,m} \right) \\ &\leq \sum_i \left(n^{1-\varepsilon/2} + \sum_{m < n^{\varepsilon/2}} r^{1-\varepsilon} \right) \\ &= |\Omega_0| \left(n^{1-\varepsilon/2} + n^{\varepsilon/2} r^{1-\varepsilon} \right) \\ &\leq C \cdot n^{1-\varepsilon/2} \end{aligned}$$

Therefore,

$$\begin{aligned} \sum_{i,m} r_{i,m} \log r_{i,m} &\geq \sum_{(i,m) \in I} r_{i,m} \log r_{i,m} \\ &\geq (1-\varepsilon) \log r \sum_{(i,m) \in I} r_{i,m} \\ &= (1-\varepsilon) \log r \cdot (r - n^{1-\varepsilon/2}) \\ &= (1 - \varepsilon - o(1)) r \log r \end{aligned}$$

In the last line we used $r \geq n/(\log n)^2 \gg n^{1-\varepsilon/2}$ □

Combining the last few lemmas, we have **proved asymptotic optimality of the Lempel-Ziv algorithm**.

Remark: There is a stronger version of the SM theorem, called the Shannon-McMillan-Breiman, which states that

$$\lim_{n \to \infty} -\frac{1}{n} \log p(x_1^n) = h \qquad a.s.$$

(the SM theorem says this happens in probability). Using this the same argument shows that that $\frac{1}{n}|c(x_1^n)| \to h$ a.s.

Remark: It is also possible to analyze the behavior of the algorithm on an individual sequence $x \in \Omega_0^{\mathbb{N}}$ and show that it is asymptotically optimal among all finite-state compression algorithms. See LZ78

Remark: We described an algorithm which searches all the way "back in time" for instances of a word. In practice one wants to limit the amount of data which is stored. In many implementation a "sliding block window" is used so as not to store all the previous data, and instead we look for repetitions only some number of symbolic into the past. This means that there will be some repetitions among the z^i but it still performs well in practice.