Universal Algorithms in Signal Processing and Communications

By:

Denver Greene

Universal Algorithms in Signal Processing and Communications

By:

Denver Greene

Online:
< http://cnx.org/content/col11524/1.1/ >

CONNEXIONS

Rice University, Houston, Texas

Collection structure revised: May 16, 2013
PDF generated: May 21, 2013
For copyright and attribution information for the modules contained in this collection, see p. 52.

Table of Contents

Chapter 1

Introduction[1]

1.1 Motivation

This course focuses on algorithms in signal processing and communications. When training data is available in such systems, we can process the data by first training on historical data and then running a Bayesian scheme, which relies on the statistics being known. A similar Bayesian approach can also be used when the statistics are approximately known.

For example, consider lossy image compression. It is well known that wavelet coefficients of images have a distribution that resembles Laplace,

$$f_x\left(x\right) = c_1 x e^{-c_2|x|}, \tag{1.1}$$

and the coefficients are approximately independent and identically distributed (i.i.d.). A well-known approach to lossy image compression is to first compute the wavelet coefficients and then compress them using a lossy compressor that is designed for Laplace i.i.d. coefficients [34]. In this approach, the training consists of the body of art that realizes that wavelet coefficients are approximately Laplace i.i.d., and the Bayesan algorithm is a lossy compressor that is designed for this distribution.

However, sometimes the statistics of the input (often called the *source*) are completely unknown, there is no training data, or there is great uncertainty in the statistics. For example, in lossless data compression, we do not know whether a file is an executable, source code, a DNA sequence, contains financial transactions, is text, etc. And even if we know that the file contains text, it has been noted that even different chapters that appear in the same book that is written by the same author may contain different statistics.

For this latter set of problems, the Bayesian approach is useless, because there is no training data. A good alternative approach to Bayesian algorithms is to use *universal algorithms* [55], [40]. These algorithms have good performance irrespective of statistics. In fact, in some cases, these algorithms can achieve (with equality) the theoretically optimum performance in an appropriate asymptotic framework.

In *lossless compression*, universal algorithms have had great impact. For example, the Lempel-Ziv family of algorithms [55], [57] asymptotically achieve the entropy rate [20], [12], which is the best possible compression ratio achievable in lossless compression, despite not knowing the input statistics. Additionally, the Lempel-Ziv algorithms allow efficient implementation.

1.2 Overview

The goal of this course is to study universal algorithms, starting from the well-trodden material in lossless compression, and later discussing universal algorithms in other areas of communications and signal processing. Let us overview the material studied during the course. We begin in Chapter 2 with a review of some

[1] This content is available online at <http://cnx.org/content/m46221/1.3/>.

information theory material, including typical sequences and source coding, in order to provide sufficient background. Next, Chapter 3 statistical models for data will be described. Chapter 4 then presents techniques for universal lossless compression of parametric sources. One approach to universal compression of parametric sources is minimum description length, which compresses the data in order to minimize for the sum of the complexity of the model and the complexity of the data given the parameters of the model. Minimum description length has been used with context tree models to provide universal contextual prediction; context tree approaches are detailed in Chapter 5. We then switch gears in Chapter 5 and move beyond lossless compression; universal lossy compression and signal reconstruction are described in detail. Finally, Chapter 7 describes Lempel-Ziv algorithms for universal lossless compression based on parsing an input sequence. For convenienve, notation is summarized in Chapter 8.

This manuscript is a work in progress, and we expect to expand and improve it during future teachings of the course.

Chapter 2

Background[1]

2.1 Convergence of random variables

We include some background material for the course. Let us recall some notions of convergence of random variables (RV's).

- A sequence of RV's $\{x_n\}_{n\geq 1}$ converges in probability if $\forall \varepsilon \geq 0, lim_{n\rightarrow\infty} supPr\left(|x_n - \overline{x}| > \varepsilon\right) = 0$. We denote this by $x_n \overset{P.}{\rightarrow} \overline{x}$.
- A sequence of RV's $\{x_n\}_{n\geq 1}$ converges to $\overline{x}$ with probability 1 if $Pr\{x_1, x_2, ... : lim_{n\rightarrow\infty} x_n = \overline{x}\} = 1$. We denote this by $x_n \overset{w.p.1}{\rightarrow} \overline{x}$ or $x_n \overset{a.s.}{\rightarrow} \overline{x}$.
- A sequence of RV's $\{x_n\}_{n\geq 1}$ converges to $\overline{x}$ in the ℓ_p sense if $E\left[|x_n - \overline{x}|^p\right] \rightarrow 0$. We denote this by $x_n \overset{\ell_p}{\rightarrow} \overline{x}$.

For example, for $p = 2$ we have mean square convergence, $x_n \overset{m.s.}{\rightarrow} \overline{x}$. For $p \geq 2$,

$$E|x_n, -, \overline{x}|^{p-1} = E(|x_n, -, \overline{x}|^p)^{\frac{p-1}{p}} \leq (E|x_n - \overline{x}|^p)^{\frac{p-1}{p}}. \tag{2.1}$$

Therefore, $x_n \overset{\ell_p}{\rightarrow} \overline{x}$ yields $x_n \overset{\ell_{p-1}}{\rightarrow} \overline{x}$. Note that for convergence in ℓ_1 sense, we have

$$Pr\left(|x_n - \overline{x}| > \varepsilon\right) \leq \frac{E|x_n - \overline{x}|}{\varepsilon} \rightarrow 0. \tag{2.2}$$

2.2 Typical Sequences

The following material appears in most textbooks on information theory (c.f., Cover and Thomas [13] and references therein). We include the highlights in order to make these notes self contained, but skip some details and the proofs. Consider a sequence $x = x^n = (x_1, x_2, ..., x_n)$, where $x_i \in \alpha$, α is the alphabet, and the cardinality of α is r, i.e., $|\alpha| = r$.

Definition 1 The *type* of x consists of the empirical probabilities of symbols in x,

$$P_x(a) = \frac{n_x(a)}{n}, \quad a \in \alpha, \tag{2.3}$$

where $n_x(a)$ is the *empirical symbol count*, which is the the number of times that $a \in \alpha$ appears in x.

Definition 2 The set of *all possible types* is defined as P_n.

[1] This content is available online at <http://cnx.org/content/m46240/1.3/>.

Example 2.1

For an alphabet $\alpha = \{0,1\}$ we have $P_n = \{\left(\frac{0}{n}, \frac{n}{n}\right), \left(\frac{1}{n}, \frac{n-1}{n}\right), ..., \left(\frac{n}{n}, \frac{0}{n}\right)\}$. In this case, $|P_n| = n+1$.

Definition 3 A *type class* T_x contains all $x^{'} \in \alpha^n$, such that $P_x {=} P_{x^{'}}$,

$$T_x = T\left(P_x\right) = \{x^{'} \in \alpha^n : P_{x^{'}} = P_x\}. \tag{2.4}$$

Example 2.2

Consider $\alpha = 1,2,3$ and $x = 11321$. We have $n = 5$ and the empirical counts are $n_x = (3,1,1)$. Therefore, the type is $P_x = \left(\frac{3}{5}, \frac{1}{5}, \frac{1}{5}\right)$, and the type class T_x contains all length-5 sequences with 3 ones, 1 two, and 1 three. That is, $T_x = \{11123, 11132, ..., 32111\}$. It is easy to see that $|T_x| = \frac{5!}{3!1!1!} = 20$.

Theorem 1 The cardinality of the set of all types satisfies $|P_n| \leq (n+1)^{r-1}$.

The proof is simple, and was given in class. We note in passing that this bound is loose, but it is good enough for our discussion.

Next, consider an i.i.d. source with the following prior,

$$Q\left(x\right) = \prod_{i=1}^{n} Q\left(x_i\right). \tag{2.5}$$

We note in passing that i.i.d. sources are sometimes called memoryless. Let the entropy be

$$H\left(P_x\right) = -\Sigma_{a\in\alpha} \frac{n_x\left(a\right)}{n} log\left(\frac{n_x\left(a\right)}{n}\right), \tag{2.6}$$

where we use base-two logarithms throughout. We are studying the entropy $H\left(P_x\right)$ in order to show that it is the fundamental performance limit in lossless compression. Σ find me

We also define the divergence as

$$D\left(P_x \parallel Q_x\right) = \Sigma_{a\in\alpha} P_x log\left(\frac{P_x}{Q_x}\right). \tag{2.7}$$

It is well known that the divergence is non-negative,

$$D\left(P_x \parallel Q_x\right) \geq 0. \tag{2.8}$$

Moreover, $D\left(P \parallel Q\right) = 0$ only if the distributions are identical.

Claim 1 The following relation holds,

$$Q\left(x\right) = 2^{-n\left[H\left(P_x\right) + D\left(P_x \| Q(x)\right)\right]}. \tag{2.9}$$

The derivation is straightforward,

$$\begin{aligned} Q\left(x\right) &= \Pi_{a\in\alpha} Q(a)^{n_x(a)} \\ &= 2^{\Sigma_{a\in\alpha} n_x(a) log Q(a)} \\ &= 2^{n\Sigma P_x(a)\left(log\frac{Q}{P} + log P\right)} \\ &= 2^{-n\left[H\left(P_x\right) + D\left(P_x \| Q(x)\right)\right]}. \end{aligned} \tag{2.10}$$

Seeing that the divergence is non-negative (2.8), and zero only if the distributions are equal, we have $Q\left(x\right) \leq P_x\left(x\right)$. When $P_x = Q$ the divergence between them is zero, and we have that $P_x\left(x\right) = Q_x = 2^{-nH\left(P_x\right)}$.

The proof of the following theorem was discussed in class.

Theorem 2 The cardinality of the type class $T\left(P_x\right)$ obeys,

$$(n+1)^{-(r-1)} \cdot 2^{nH\left(P_x\right)} \leq |T\left(P_x\right)| \leq 2^{nH\left(P_x\right)}. \tag{2.11}$$

Having computed the probability of x and cardinality of its type class, we can easily compute the probability of the type class.

Claim 2 The probability $Q\left(T\left(P_x\right)\right.$ of the type class $T\left(P_x\right)$ obeys,

$$(n+1)^{-(r-1)} \cdot 2^{-nD(P_x\|Q_x)} \leq Q\left(T\left(P_x\right)\right) \leq 2^{-nD(P_x\|Q_x)}. \tag{2.12}$$

Consider now an event A that is a union over $T\left(P_x\right)$. Suppose $T\left(Q\right) \nsubseteq A$, then A is rare with respect to (w.r.t) the prior Q. and we have $lim_{n\to\infty} Q\left(A\right) = 0$. That is, the probability is concentrated around Q. In general, the probability assigned by the prior Q to an event A satisfies

$$\begin{aligned} Q\left(A\right) &= \Sigma_{x\in A} Q\left(x\right) = \Sigma_{T(P_x)\subseteq A} Q\left(T\left(P_x\right)\right) \\ &\doteq \Sigma_{T(P_x)\subseteq A} 2^{-nD(P_x\|Q)} \\ &\doteq 2^{-n\cdot min_{p\in A} D(P\|Q)}, \end{aligned} \tag{2.13}$$

where we denote $a_n \doteq b_n$ when $\frac{1}{n} log\left(\frac{a_n}{b_n}\right) \to 0$.

2.3 Fixed and Variable Length Coding

Fixed to fixed length source coding: As before, we have a sequence x of length n, and each element of x is from the alphabet α. A *source code* maps the input $x^n \in r^n$ to a set of 2^{Rn} bit vectors, each of length Rn. The rate R quantifies the number of output bits of the code per input element of x.[2] That is, the output of the code consists of nR bits. If n and R is fixed, then we call this a *fixed to fixed* length source code.

The decoder processes the nR bits and yields $\overset{\Theta}{x} \in \alpha^n$. Ideally we have that $\overset{\Theta}{x} = x$, but if $2^{nR} < r^n$ then there are inputs that are not mapped to any output, and $\overset{\Theta}{x}$ may differ from x. Therefore, we want $Pr\left(\overset{\Theta}{x} \neq x\right)$ to be small. If R is too small, then the error probability will go to 1. On the other hand, sufficiently large R will drive this error probability to 0 as n is increased.

If $log\left(r\right) > R$ and $Pr\left(\overset{\Theta}{x} \neq x\right)$ is vanishing as n is increased, then we are compressing, because $2^{log(r)n} = r^n > 2^{Rn}$, where r^n is the number of possible inputs x and there are 2^{Rn} possible outputs.

What is a good fixed to fixed length source code? One option is to map $2^{Rn} - 1$ outputs to inputs with high probabilities, and the last output can be mapped to a "don't care" input. We will discuss the performance of this style of code.

An input $x \in r^n$ is called δ-typical if $Q\left(x\right) > 2^{-(H+\delta)n}$. We denote the set of δ-typical inputs by $T_Q\left(\delta\right)$, this set includes the type classes whose empirical probabilities are equal (or closest) to the true prior $Q\left(x\right)$. Note that for each type class T_x, all inputs $x^{'} \in T_x$ in the type class have the same probability, i.e., $Q\left(x^{'}\right) = Q\left(x\right)$. Therefore, the set $T_Q\left(\delta\right)$ is a union of type classes, and can be thought of as an event A (Section 2.2 (Typical Sequences)) that contains type classes consisting of high-probability sequences. It is easily seen that the event A contains the true i.i.d. distribution Q, because sequences whose empirical probabilities satisfy $P_x = Q$ also satisfy

$$Q\left(x\right) = 2^{-Hn} > 2^{-(H+\delta)n}. \tag{2.14}$$

Using the principles discussed in Section 2.2 (Typical Sequences), it is readily seen that the probability under the prior Q of the inputs in $T_Q\left(\delta\right)$ satisfies $Q\left(T_p\left(\delta\right)\right) = Q\left(A\right) \to 1$ when $n \to \infty$. Therefore, a code $\mathcal{C}$ that enumerates $T_Q\left(\delta\right)$ will encode x correctly with high probability.

The key question is the size of $\mathcal{C}$, or the cardinality of $T_Q\left(\delta\right)$. Because each $x \in T_Q\left(\delta\right)$ satisfies $Q\left(x\right) > 2^{(-H+\delta)n}$, and $\sum_{x\in T_Q(\delta)} Q\left(x\right) \leq 1$, we have $|T_Q\left(\delta\right)| < 2^{(H+\delta)n}$. Therefore, a rate $R \geq H + \delta$ allows *near-lossless coding*, because the probability of error vanishes (recall that $Q\left(\left(T_p\left(\delta\right)\right)^C\right) \to 0$, where $\left(\cdot\right)^C$ denotes the complement).

[2] We assume without loss of generality that $Rn \in \mathbb{Z}$. If not, then we can round Rn up to $\lceil Rn \rceil$, where $\lceil\cdot\rceil$ denotes rounding up.

On the other hand, a rate $R \leq H - \delta$ will not allow lossless coding, and the probability of error will go to 1. We will see this intuitively. Because the type class whose empirical probability is Q dominates, a type class T_x whose sequences have larger probability, e.g., $Q(x) > 2^{-(H-\delta)n}$, will have small probability in aggregate. That is,

$$\sum_{x:Q(x)>2^{-n(H-\delta)}} Q(x) \overset{n\to\infty}{\to} 0. \tag{2.15}$$

In words, choosing a code $\mathcal{C}$ with rate $R = H - \delta$ that contains the words x with highest probability will fail, it will not cover enough probabilistic mass. We conclude that near-lossless coding is possible at rates above H and impossible below H.

To see things from a more intuitive angle, consider the definition of entropy, $H(Q) = -\sum_{a\in\alpha} Q(a)\, log(Q(a))$. If we consider each bit as reducing uncertainty by a factor of 2, then the average log-likelihood of a length-n input x generated by Q satisfies

$$\begin{array}{rcc} E[-log(Pr(x))] & = & E\left[-log\left(\prod_{i=1}^{n} Pr(x_i)\right)\right] \\ & = & -\sum_{i=1}^{n} E[log(Q(x_i))] \\ & = & -\sum_{i=1}^{n} \sum_{a\in\alpha} Q(a)\cdot log(Q(a)) \\ & = & nH. \end{array} \tag{2.16}$$

Because the expected log-likelihood of x is nH, it will take nH bits to reduce the uncertainty by this factor.

Fixed to variable length source coding: The near-lossless coding above relies on enumerating a collection of high-probability codewords $T_Q(\delta)$. However, this approach suffers from a troubling failure for $x \notin T_Q(\delta)$. To solve this problem, we incorporate a code that maps x to an output consisting of a *variable* number of bits. That is, the length of the code will be approximately nH on average, but could be greater or lesser.

One possible variable length code is due to Shannon. Consider all possible $x \in \alpha^n$. For each x, allocate $\lceil -log(Q(x)) \rceil$ bits to x. It can be shown that it is possible to construct an invertible (uniquely decodable) code as long as the length of the code $l(x)$ in bits allocated to each x satisfies

$$\sum_x 2^{-l(x)} \leq 1. \tag{2.17}$$

This result is known as the Kraft Inequality. Seeing that

$$\begin{array}{rcl} \sum_x 2^{-l(x)} & = & \sum_x 2^{-\lceil -log(Q(x)) \rceil} \\ & \leq & \sum_x 2^{-(-log(Q(x)))} \\ & = & \sum_x Q(x) = 1, \end{array} \tag{2.18}$$

we see that the length allocation we suggested satisfies the Kraft Inequality. Therefore, it is possible to construct an invertible (and hence lossless) code with lengths upper bounded by

$$l_x = \lceil -log(Q(x)) \rceil \leq -log(Q(x)) + 1, \tag{2.19}$$

and we have

$$E[l(x)] \leq E[-log(Q(x))] + 1 = nH + 1. \tag{2.20}$$

This simple construction approaches the entropy up to 1 bit.

Unfortunately, a Shannon code is impractical, because it requires to construct a code book of exponential size $|\alpha|^n$. Instead, arithmetic codes [42] are used; we discussed arithmetic codes in detail in class, but they appear in all standard text books and so we do not describe them here.

Chapter 3

Source models[1]

For i.i.d. sources, $D\left(P_1\left(x^n\right)||P_2\left(x^n\right)\right) = nD\left(P_1\left(x_i\right)||P_2\left(x_i\right)\right)$, which means that the divergence increases linearly with n. Not only does the divergence increase, but it does so by a constant per symbol. Therefore, based on typical sequence concepts that we have seen, for an x^n generated by P_1, its probability under P_2 vanishes. However, we can construct a distribution Q whose divergence with both P_1 abd P_2 is small,

$$Q\left(x^n\right) = \frac{1}{2}P_1\left(x^n\right) + \frac{1}{2}P_2\left(x^n\right). \tag{3.1}$$

We now have for P_1,

$$\begin{array}{rcl} \frac{1}{n}D\left(P_1^n||Q\right) & = & \frac{1}{n}E\left[log\frac{P_1(x^n)}{\frac{1}{2}P_1(x^n)+\frac{1}{2}P_2(x^n)}\right] \\ & \leq & \frac{1}{n}log\left(2\right) = \frac{1}{n}. \end{array} \tag{3.2}$$

On the other hand, $\frac{1}{n}D\left(P_1\left(x_1^n\right)||Q\left(x_1^n\right)\right) \geq 0$ (2.8), and so

$$\frac{1}{n} \geq \frac{1}{n}D\left(P_1\left(x_1^n\right)||Q\left(x_1^n\right)\right) \geq 0. \tag{3.3}$$

By symmetry, we see that Q is also close to P_2 in the divergence sense.

Intuitively, it might seem peculiar that Q is close to both P_1 and P_2 but they are far away from each other (in divergence terms). This intuition stems from the triangle inequality, which holds for all metrics. The contradiction is resolved by realizing that the divergence is not a metric, and it does not satisfy the triangle inequality.

Note also that for two i.i.d. distributions P_1 and P_2, the divergence

$$D\left(P_1\left(x^n\right)||P_2\left(x^n\right)\right) = nD\left(P_1||P_2\right) \tag{3.4}$$

is linear in n. If Q were i.i.d., then $D\left(P_1\left(x^n\right) \parallel Q\left(x_1^n\right)\right)$ must also be linear in n. But the divergence is not increasing linearly in n, it is upper bounded by 1. Therefore, we conclude that $Q\left(\cdot\right)$ is not an i.i.d. distribution. Instead, Q is a distribution that contains memory, and there is dependence in Q between collections of different symbols of x in the sense that they are either all drawn from P_1 or all drawn from P_2. To take this one step further, consider K sources with

$$Q\left(x^n\right) = \sum_{i=1}^{K}\frac{1}{K}P_i\left(x^n\right), \tag{3.5}$$

[1] This content is available online at <http://cnx.org/content/m46231/1.4/>.

then in an analogous manner to before it can be shown that

$$D\left(P_i\left(x_1^n\right)||Q\left(x_1^n\right)\right) \le \frac{1}{n} log\left(K\right). \tag{3.6}$$

Sources with memory: Instead of the memoryless (i.i.d.) source,

$$P\left(x^n\right) = \prod_{i=1}^{n} P\left(x_i\right), \tag{3.7}$$

let us now put forward a statistical model with memory,

$$P\left(x^n\right) = \prod_{i=1}^{n} P\left(x_i | x_1^{i-1}\right). \tag{3.8}$$

Stationary source: To understand the notion of a stationary source, consider an infinite stream of symbols, $..., x_{-1}, x_0, x_1, ...$ A complete probabilistic description of a stationary distribution is given by the collection of all marginal distribution of the following form for all t and n,

$$P_{X_t, X_{t+1}, ..., X_{t+n-1}}\left(x_t, x_{t+1}, ..., x_{t+n-1}\right). \tag{3.9}$$

For a stationary source, this distribution is independent of t.

Entropy rate: We have defined the first order entropy of an i.i.d. random variable (2.6), and let us discuss more advanced concepts for sources with memory. Such definitions appear in many standard textbooks, for example that by Gallager [21].

1. The *order-n entropy* is defined,

$$H_n = \frac{1}{n} H\left(x_1, ..., x_n\right) = -\frac{1}{n} E\left[log\left(P\left(x_1, ..., x_n\right)\right)\right]. \tag{3.10}$$

2. The *entropy rate* is the limit of order-n entropy, $\overline{H} = lim_{n\to\infty} H_n$. The existence of this limit will be shown soon.
3. *Conditional entropy* is defined similarly to entropy as the expectation of the log of the conditional probability,

$$H\left(x_n | x_1, ..., x_{n-1}\right) = -\frac{1}{n} E\left[log\left(P\left(x_n | x_1, ..., x_{n-1}\right)\right)\right], \tag{3.11}$$

where expectation is taken over the joint probability space, $P\left(x_1, ..., x_n\right)$.

The entropy rate also satisfies $\overline{H} = lim_{n\to\infty} H\left(x_n | x_1, ..., x_n\right)$.

Theorem 3 For a stationary source with bounded first order entropy, $H_1\left(x\right) < \infty$, the following hold.

1. The conditional entropy $H\left(x_n | x_1, ..., x_{n-1}\right)$ is monotone non-increasing in n.
2. The order-n entropy is not smaller than the conditional entropy,

$$H_n\left(x\right) \ge H\left(x_n | x_1, ..., x_{n-1}\right). \tag{3.12}$$

3. The order-n entropy $H_n\left(x\right)$ is monotone non-increasing.
4. $\overline{H}\left(x\right) = lim_{n\to\infty} H_n\left(x\right) = lim_{n\to\infty} H\left(x_n | x_1, ..., x_{n-1}\right)$.

Proof. **Part (1):**

$$\begin{aligned} H\left(x_n | x_1, ..., x_{n-1}\right) \quad & \le H\left(x_n | x_2, ..., x_{n-1}\right) \\ & = H\left(x_{n-1} | x_1, ..., x_{n-2}\right) \\ & \le ... \le H\left(x_2 | x_1\right) \le H\left(x_1\right). \end{aligned} \tag{3.13}$$

Part (2):

$$\begin{array}{ll} H_n\left(x\right) & = \frac{1}{n}\left[H\left(x_1\right) + H\left(x_2|x_1\right) + ... + H\left(x_n|x_1,...,x_{n-1}\right)\right] \\ & \geq \frac{1}{n}\left[H\left(x_n|x_1,...,x_{n-1}\right) + ... + H\left(x_n|x_1,...,x_{n-1}\right)\right] \\ & = H\left(x_n|x_1,...,x_{n-1}\right). \end{array} \tag{3.14}$$

Part (3): This comes from the fist equality in the proof of (2), because we have the average of a monotonely non-increasing sequence.

Part (4): Both sequences are monotone non-increasing (parts (1) and (3)) and bounded below (by zero). Therefore, they both have a limit. Denote $\overline{H}\left(x\right) = lim_{n\rightarrow\infty}H_n\left(x\right)$ and $\tilde{H}\left(x\right) = lim_{n\rightarrow\infty}H\left(x_n|x_1,...,x_{n-1}\right)$.

Owing to part(2), $\overline{H} \geq \tilde{H}$. Therefore, it suffices to prove $\tilde{H} \geq \overline{H}$.

$$\begin{array}{ll} H_{n+m}\left(x\right) & = \frac{1}{n+m}\left[H\left(x_1^{n-1}\right) + \sum_{i=n}^{n+m} H\left(x_i|x_1,...,x_{i-1}\right)\right] \\ & \leq \frac{H\left(x_1^{n-1}\right)}{n+m} + \frac{m+1}{n+m}H\left(x_i|x_1,...,x_{i-1}\right). \end{array} \tag{3.15}$$

Now fix n and take the limit for large m. The inequality $\overline{H} \leq \tilde{H}$ appears, which proves that both limits are equal.

Coding theorem: Theorem 3 (p. 8) yields for fixed to variable length coding that for a stationary source, there exists a lossless code such that the compression rate ρ_n obeys,

$$\rho_n = \frac{E\left[l\left(x_1,...,x_n\right)\right]}{n} \leq H_n\left(x\right) + \frac{1}{n}. \tag{3.16}$$

This can be proved, for example, by choosing $l\left(x_1,...,x_n\right) = \lceil -logP\left(x_1,...,x_n\right)\rceil$, which is a Shannon code. As n is increased, the compression rate ρ_n converges to the entropy rate.

We also have a converse theorem for lossless coding of stationary sources. That is, $\rho_n \geq H_n\left(x\right) \geq \overline{H}$.

3.1 Stationary Ergodic Sources

Consider the sequence $x = \left(..., x_{-1}, x_0, x_1, ...\right)$. Let $x^{'} = S_x$ denote a step $\forall n \in \mathbb{Z}$, $x_n^{'} = x_{n+1}$, where S_x^i takes i steps. Let $f_k\left(x\right)$ be a function that operates on coordinates $\left(x_0, ..., x_{k-1}\right)$. An ergodic source has the property that empirical averages converge to statistical averages,

$$\frac{1}{n}\sum_{i=0}^{n-1} f_k\left(S_x^i\right) \overset{a.s.,n\rightarrow\infty}{\rightarrow} Ef_k\left(x\right). \tag{3.17}$$

In block codes we want

$$\frac{1}{nN}\sum_{i=0}^{n-1} l\left(x_{iN+1}, ..., x_{(i+1)N}\right) \overset{\text{a.s.}}{=} H. \tag{3.18}$$

We will be content with convergence in probability, and a.s. convergence is better.

Theorem 4 Let X be a stationary ergodic source with $H_1\left(x\right) < \infty$, then for every $\varepsilon > 0, \delta > 0$, there exists $n_0\left(\delta, \varepsilon\right)$ such that $\forall n \geq n_0\left(\delta, \varepsilon\right)$,

$$Pr\{|\frac{1}{n}I\left(x_1, ..., x_n\right) - \overline{H}|\} \leq \varepsilon, \tag{3.19}$$

where $I\left(x_1, ..., x_n\right) = -log\left(Pr\left(x_1, ..., x_n\right)\right)$.

The proof of this result is quite lengthy. We discussed it in detail, but skip it here.

Theorem 4 (p. 9) is called the ergodic theorem of information theory or the ergodic theorem of entropy. Shannon (48') proved convergence in probability for stationary ergodic Markov sources. McMillan (53') proved L^1 convergence for stationary ergodic sources. Brieman (57'/60') proved convergence with probability 1 for stationary ergodic sources.

3.2 Parametric Models of Information Sources

In this section, we will discuss several parametric models and see what their entropy rate is.

Memoryless sources: We have seen for memoryless sources,

$$p(x) = \prod_{i=1}^{n} p(x_i), \tag{3.20}$$

where there are $r-1$ parameters in total,

$$\theta = \{p(a), a = 1, 2, ..., r-1\}, \tag{3.21}$$

the parameters are denoted by θ, and $\alpha = \{1, 2, ..., r\}$ is the alphabet.

Markov sources: The distribution of a Markov source is defined as

$$p(x_1, x_2, ..., x_n) = p(x_1, ..., x_k) \prod_{i=k+1}^{n} p\left(x_i | x_{i-k}^{i-1}\right), \tag{3.22}$$

where $n \geq k$. We must define $\{p(a_1, a_2, ..., a_k)\}_{(a_1, a_2, ... a_k) \in \alpha^k}$ initial probabilities and transition probabilities, $\{p\left(a_{k+1} | a_1^k\right)\}$. There are $r^k - 1$ initial probabilities and $(r-1) r^k$ transition probabilities, giving a total of $r^{k+1} - 1$ parameters. Note that

$$E\{-logp\left(x_i | x_{i-k}^{i-1}\right)\} = H\left(X_i | X_{i-k}^{i-1}\right) \overset{k \to \infty}{\to} \overline{H}(X). \tag{3.23}$$

Therefore, the space of Markov sources covers the stationary ergodic sources in the limit of large k.

Unifilar sources: For unifilar sources, it is possible to reconstruct the set of states that a source went through by looking at the output sequence. In the Markov case we have $S_i = \left(X_{i-k}^{i-1}\right)$, but in general it may be more complicated to determine the state.

To put us on a concrete basis for analysis of unifilar sources, consider a source with M states, $S = \{1, 2, ..., M\}$, and an alphabet $\alpha = \{1, 2, ..., r\}$. In each time step, the source outputs a symbol and moves to a new state. Denote the output sequence by $x = x_1 x_2 \cdots x_n$, and the state sequence by $s = s_1 s_2 \cdots s_n$, where $s_i \in S$ and $x_i \in \alpha$. Denote also

$$\begin{aligned} q\left(s | s^{'}\right) &= \Pr\{S_t = s | S_{t-1} = s^{'}\} \\ &= \Pr\{S_t = s | S_{t-1} = s^{'}, S_{t-2}, \cdots\}. \end{aligned} \tag{3.24}$$

This is a first-order time-homogeneous Markov source. The probability that the next symbol is a follows,

$$\begin{aligned} p(a|s) &= \Pr\{X_t = a | S_t = s\} \\ &= \Pr\{X_t = a | S_t = s, X_{t-1}, S_{t-1}, \cdots\}. \end{aligned} \tag{3.25}$$

There exists a deterministic function,

$$S_t = g(S_{t-1}, X_{t-1}), \tag{3.26}$$

this is called the next state function. Given that we start at some state $S_1 = s_1$, the probability for the sequence of states $s_1, ..., s_n$ is given by

$$p\left(X_1^n | S_1\right) = \prod_{t=1}^{n} p(X_t | S_t). \tag{3.27}$$

Note the relation

$$q\left(s | s^{'}\right) = \sum_{a : g(s^{'}, a) = s} p\left(a | s^{'}\right). \tag{3.28}$$

To summarize, unifilar sources can be described by a state machine style of diagram as illustrated in Figure 3.1.

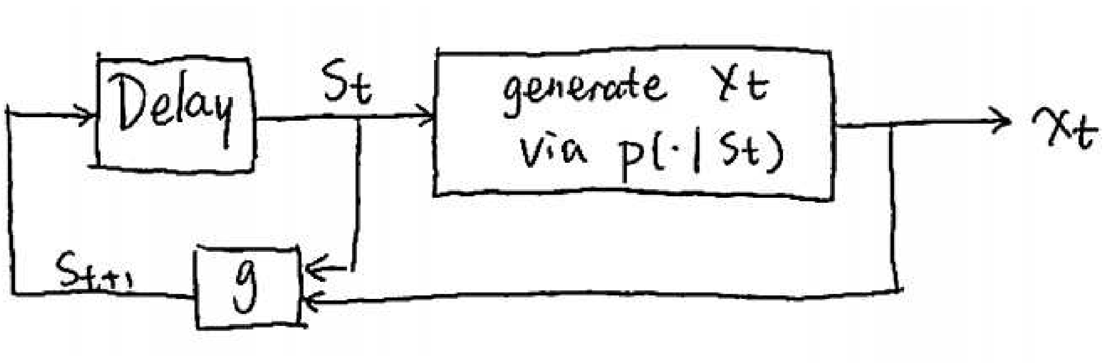

Figure 3.1: State machine for selecting the state of a unifilar source.

Given that an initial state was fixed, a unifilar source with M states and an alphabet of size r can be expressed with $M(r-1)$ parameters. If the initial state is a random variable, then there are $M-1$ parameters that define probabilities for the initial state, giving $M(r-1)+M-1=Mr-1$ parameters in total. In the Markov case, we have $M=r^k$, it is a special type of unifilar source.

Example 3.1

For the unifilar source that appears in Figure 3.2, the states can be discerned from the output sequence. Let us follow up on this example while discussing more properties of unifilar sources.