theorem states that the first few elements are asymptotically independent with common distribution P^*.

Example 11.6.2 As an example of the conditional limit theorem, let us consider the case when n fair dice are rolled. Suppose that the sum of the outcomes exceeds $4n$. Then by the conditional limit theorem, the probability that the first die shows a number $a \in \{1, 2, \ldots, 6\}$ is approximately $P^*(a)$, where $P^*(a)$ is the distribution in E that is closest to the uniform distribution, where $E = \{P : \sum P(a)a \geq 4\}$. This is the maximum entropy distribution given by

$$P^*(x) = \frac{2^{\lambda x}}{\sum_{i=1}^{6} 2^{\lambda i}}, \tag{11.173}$$

with λ chosen so that $\sum i P^*(i) = 4$ (see Chapter 12). Here P^* is the conditional distribution on the first (or any other) die. Apparently, the first few dice inspected will behave as if they are drawn independently according to an exponential distribution.

11.7 HYPOTHESIS TESTING

One of the standard problems in statistics is to decide between two alternative explanations for the data observed. For example, in medical testing, one may wish to test whether or not a new drug is effective. Similarly, a sequence of coin tosses may reveal whether or not the coin is biased.

These problems are examples of the general hypothesis-testing problem. In the simplest case, we have to decide between two i.i.d. distributions. The general problem can be stated as follows:

Problem 11.7.1 Let $X_1, X_2, \ldots, X_n$ be i.i.d. $\sim Q(x)$. We consider two hypotheses:

- H_1: $Q = P_1$.
- H_2: $Q = P_2$.

Consider the general decision function $g(x_1, x_2, \ldots, x_n)$, where $g(x_1, x_2, \ldots, x_n) = 1$ means that H_1 is accepted and $g(x_1, x_2, \ldots, x_n) = 2$ means that H_2 is accepted. Since the function takes on only two values, the test can also be specified by specifying the set A over which $g(x_1, x_2, \ldots, x_n)$ is 1; the complement of this set is the set where $g(x_1, x_2, \ldots, x_n)$ has the value 2. We define the two probabilities of error:

$$\alpha = \Pr(g(X_1, X_2, \ldots, X_n) = 2 | H_1 \text{ true}) = P_1^n(A^c) \tag{11.174}$$

and

$$\beta = \Pr(g(X_1, X_2, \ldots, X_n) = 1 | H_2 \text{ true}) = P_2^n(A). \tag{11.175}$$

In general, we wish to minimize both probabilities, but there is a trade-off. Thus, we minimize one of the probabilities of error subject to a constraint on the other probability of error. The best achievable error exponent in the probability of error for this problem is given by the Chernoff–Stein lemma.

We first prove the Neyman–Pearson lemma, which derives the form of the optimum test between two hypotheses. We derive the result for discrete distributions; the same results can be derived for continuous distributions as well.

Theorem 11.7.1 (*Neyman–Pearson lemma*) *Let $X_1, X_2, \ldots, X_n$ be drawn i.i.d. according to probability mass function Q. Consider the decision problem corresponding to hypotheses $Q = P_1$ vs. $Q = P_2$. For $T \geq 0$, define a region*

$$A_n(T) = \left\{ x^n : \frac{P_1(x_1, x_2, \ldots, x_n)}{P_2(x_1, x_2, \ldots, x_n)} > T \right\}. \tag{11.176}$$

Let

$$\alpha^* = P_1^n(A_n^c(T)), \qquad \beta^* = P_2^n(A_n(T)) \tag{11.177}$$

be the corresponding probabilities of error corresponding to decision region A_n. Let B_n be any other decision region with associated probabilities of error α and β. If $\alpha \leq \alpha^*$, then $\beta \geq \beta^*$.

Proof: Let $A = A_n(T)$ be the region defined in (11.176) and let $B \subseteq \mathcal{X}^n$ be any other acceptance region. Let ϕ_A and ϕ_B be the indicator functions of the decision regions A and B, respectively. Then for all $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathcal{X}^n$,

$$(\phi_A(\mathbf{x}) - \phi_B(\mathbf{x}))(P_1(\mathbf{x}) - T P_2(\mathbf{x})) \geq 0. \tag{11.178}$$

This can be seen by considering separately the cases $\mathbf{x} \in A$ and $\mathbf{x} \notin A$. Multiplying out and summing this over the entire space, we obtain

$$0 \leq \sum (\phi_A P_1 - T \phi_A P_2 - P_1 \phi_B + T P_2 \phi_B) \tag{11.179}$$

$$= \sum_A (P_1 - TP_2) - \sum_B (P_1 - TP_2) \tag{11.180}$$

$$= (1 - \alpha^*) - T\beta^* - (1 - \alpha) + T\beta \tag{11.181}$$

$$= T(\beta - \beta^*) - (\alpha^* - \alpha). \tag{11.182}$$

Since $T \geq 0$, we have proved the theorem. □

The Neyman–Pearson lemma indicates that the optimum test for two hypotheses is of the form

$$\frac{P_1(X_1, X_2, \ldots, X_n)}{P_2(X_1, X_2, \ldots, X_n)} > T. \tag{11.183}$$

This is the likelihood ratio test and the quantity $\frac{P_1(X_1, X_2, \ldots, X_n)}{P_2(X_1, X_2, \ldots, X_n)}$ is called the *likelihood ratio*. For example, in a test between two Gaussian distributions [i.e., between $f_1 = \mathcal{N}(1, \sigma^2)$ and $f_2 = \mathcal{N}(-1, \sigma^2)$], the likelihood ratio becomes

$$\frac{f_1(X_1, X_2, \ldots, X_n)}{f_2(X_1, X_2, \ldots, X_n)} = \frac{\prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X_i - 1)^2}{2\sigma^2}}}{\prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X_i + 1)^2}{2\sigma^2}}} \tag{11.184}$$

$$= e^{+\frac{2\sum_{i=1}^n X_i}{\sigma^2}} \tag{11.185}$$

$$= e^{+\frac{2n\overline{X}_n}{\sigma^2}}. \tag{11.186}$$

Hence, the likelihood ratio test consists of comparing the sample mean $\overline{X}_n$ with a threshold. If we want the two probabilities of error to be equal, we should set $T = 1$. This is illustrated in Figure 11.8.

In Theorem 11.7.1 we have shown that the optimum test is a likelihood ratio test. We can rewrite the log-likelihood ratio as

$$L(X_1, X_2, \ldots, X_n) = \log \frac{P_1(X_1, X_2, \ldots, X_n)}{P_2(X_1, X_2, \ldots, X_n)} \tag{11.187}$$

$$= \sum_{i=1}^n \log \frac{P_1(X_i)}{P_2(X_i)} \tag{11.188}$$

$$= \sum_{a \in \mathcal{X}} n P_{X^n}(a) \log \frac{P_1(a)}{P_2(a)} \tag{11.189}$$

$$= \sum_{a \in \mathcal{X}} n P_{X^n}(a) \log \frac{P_1(a)}{P_2(a)} \frac{P_{X^n}(a)}{P_{X^n}(a)} \tag{11.190}$$

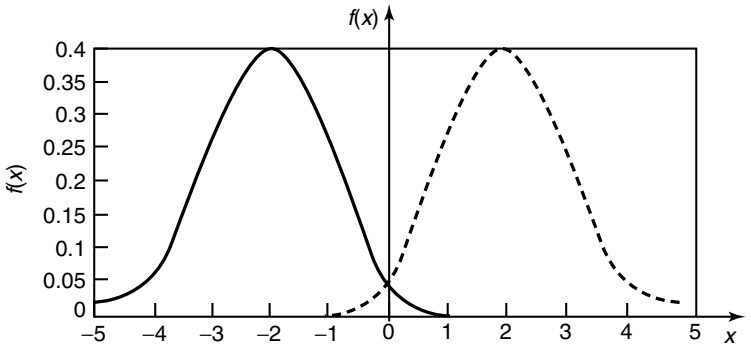

FIGURE 11.8. Testing between two Gaussian distributions.

$$= \sum_{a \in \mathcal{X}} n P_{X^n}(a) \log \frac{P_{X^n}(a)}{P_2(a)}$$

$$- \sum_{a \in \mathcal{X}} n P_{X^n}(a) \log \frac{P_{X^n}(a)}{P_1(a)} \tag{11.191}$$

$$= nD(P_{X^n} || P_2) - nD(P_{X^n} || P_1), \tag{11.192}$$

the difference between the relative entropy distances of the sample type to each of the two distributions. Hence, the likelihood ratio test

$$\frac{P_1(X_1, X_2, \ldots, X_n)}{P_2(X_1, X_2, \ldots, X_n)} > T \tag{11.193}$$

is equivalent to

$$D(P_{X^n} || P_2) - D(P_{X^n} || P_1) > \frac{1}{n} \log T. \tag{11.194}$$

We can consider the test to be equivalent to specifying a region of the simplex of types that corresponds to choosing hypothesis H_1. The optimum region is of the form (11.194), for which the boundary of the region is the set of types for which the difference between the distances is a constant. This boundary is the analog of the perpendicular bisector in Euclidean geometry. The test is illustrated in Figure 11.9.

We now offer some informal arguments based on Sanov's theorem to show how to choose the threshold to obtain different probabilities of error. Let B denote the set on which hypothesis 1 is accepted. The probability

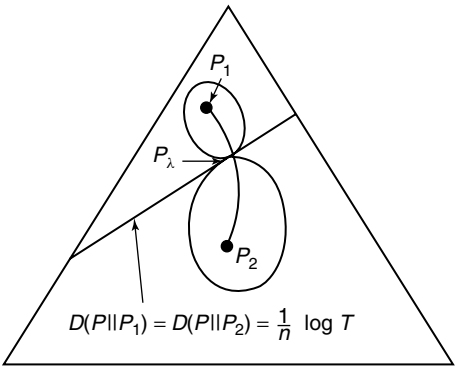

FIGURE 11.9. Likelihood ratio test on the probability simplex.

of error of the first kind is

$$\alpha_n = P_1^n(P_{X^n} \in B^c). \tag{11.195}$$

Since the set B^c is convex, we can use Sanov's theorem to show that the probability of error is determined essentially by the relative entropy of the closest member of B^c to P_1. Therefore,

$$\alpha_n \doteq 2^{-nD(P_1^*||P_1)}, \tag{11.196}$$

where P_1^* is the closest element of B^c to distribution P_1. Similarly,

$$\beta_n \doteq 2^{-nD(P_2^*||P_2)}, \tag{11.197}$$

where P_2^* is the closest element in B to the distribution P_2.

Now minimizing $D(P||P_2)$ subject to the constraint $D(P||P_2) - D(P||P_1) \geq \frac{1}{n} \log T$ will yield the type in B that is closest to P_2. Setting up the minimization of $D(P||P_2)$ subject to $D(P||P_2) - D(P||P_1) = \frac{1}{n} \log T$ using Lagrange multipliers, we have

$$J(P) = \sum P(x) \log \frac{P(x)}{P_2(x)} + \lambda \sum P(x) \log \frac{P_1(x)}{P_2(x)} + \nu \sum P(x). \tag{11.198}$$

Differentiating with respect to $P(x)$ and setting to 0, we have

$$\log \frac{P(x)}{P_2(x)} + 1 + \lambda \log \frac{P_1(x)}{P_2(x)} + \nu = 0. \tag{11.199}$$

Solving this set of equations, we obtain the minimizing P of the form

$$P_2^* = P_{\lambda^*} = \frac{P_1^\lambda(x) P_2^{1-\lambda}(x)}{\sum_{a \in \mathcal{X}} P_1^\lambda(a) P_2^{1-\lambda}(a)}, \tag{11.200}$$

where λ is chosen so that $D(P_{\lambda^*}||P_1) - D(P_{\lambda^*}||P_2) = \frac{1}{n} \log T$.

From the symmetry of expression (11.200), it is clear that $P_1^* = P_2^*$ and that the probabilities of error behave exponentially with exponents given by the relative entropies $D(P^*||P_1)$ and $D(P^*||P_2)$. Also note from the equation that as $\lambda \to 1$, $P_\lambda \to P_1$ and as $\lambda \to 0$, $P_\lambda \to P_2$. The curve that P_λ traces out as λ varies is a geodesic in the simplex. Here P_λ is a normalized convex combination, where the combination is in the exponent (Figure 11.9).

In the next section we calculate the best error exponent when one of the two types of error goes to zero arbitrarily slowly (the Chernoff–Stein lemma). We will also minimize the weighted sum of the two probabilities of error and obtain the Chernoff information bound.

11.8 CHERNOFF–STEIN LEMMA

We consider hypothesis testing in the case when one of the probabilities of error is held fixed and the other is made as small as possible. We will show that the other probability of error is exponentially small, with an exponential rate equal to the relative entropy between the two distributions. The method of proof uses a relative entropy version of the AEP.

Theorem 11.8.1 (*AEP for relative entropy*) *Let $X_1, X_2, \ldots, X_n$ be a sequence of random variables drawn i.i.d. according to $P_1(x)$, and let $P_2(x)$ be any other distribution on $\mathcal{X}$. Then*

$$\frac{1}{n} \log \frac{P_1(X_1, X_2, \ldots, X_n)}{P_2(X_1, X_2, \ldots, X_n)} \to D(P_1||P_2) \qquad \textit{in probability.} \tag{11.201}$$

Proof: This follows directly from the weak law of large numbers.

$$\frac{1}{n} \log \frac{P_1(X_1, X_2, \ldots, X_n)}{P_2(X_1, X_2, \ldots, X_n)} = \frac{1}{n} \log \frac{\prod_{i=1}^n P_1(X_i)}{\prod_{i=1}^n P_2(X_i)} \tag{11.202}$$

$$= \frac{1}{n}\sum_{i=1}^{n} \log \frac{P_1(X_i)}{P_2(X_i)} \tag{11.203}$$

$$\rightarrow E_{P_1} \log \frac{P_1(X)}{P_2(X)} \text{ in probability} \tag{11.204}$$

$$= D(P_1||P_2). \qquad \square \tag{11.205}$$

Just as for the regular AEP, we can define a relative entropy typical sequence as one for which the empirical relative entropy is close to its expected value.

Definition For a fixed n and $\epsilon > 0$, a sequence $(x_1, x_2, \ldots, x_n) \in \mathcal{X}^n$ is said to be *relative entropy typical* if and only if

$$D(P_1||P_2) - \epsilon \leq \frac{1}{n} \log \frac{P_1(x_1, x_2, \ldots, x_n)}{P_2(x_1, x_2, \ldots, x_n)} \leq D(P_1||P_2) + \epsilon. \tag{11.206}$$

The set of relative entropy typical sequences is called the *relative entropy typical set* $A_\epsilon^{(n)}(P_1||P_2)$.

As a consequence of the relative entropy AEP, we can show that the relative entropy typical set satisfies the following properties:

Theorem 11.8.2

1. *For* $(x_1, x_2, \ldots, x_n) \in A_\epsilon^{(n)}(P_1||P_2)$,

$$\begin{aligned} &P_1(x_1, x_2, \ldots, x_n)2^{-n(D(P_1||P_2)+\epsilon)} \\ &\quad \leq P_2(x_1, x_2, \ldots, x_n) \\ &\quad \leq P_1(x_1, x_2, \ldots, x_n)2^{-n(D(P_1||P_2)-\epsilon)}. \end{aligned} \tag{11.207}$$

2. $P_1(A_\epsilon^{(n)}(P_1||P_2)) > 1 - \epsilon$, *for n sufficiently large.*
3. $P_2(A_\epsilon^{(n)}(P_1||P_2)) < 2^{-n(D(P_1||P_2)-\epsilon)}$.
4. $P_2(A_\epsilon^{(n)}(P_1||P_2)) > (1 - \epsilon)2^{-n(D(P_1||P_2)+\epsilon)}$, *for n sufficiently large.*

Proof: The proof follows the same lines as the proof of Theorem 3.1.2, with the counting measure replaced by probability measure P_2. The proof of property 1 follows directly from the definition of the relative entropy

typical set. The second property follows from the AEP for relative entropy (Theorem 11.8.1). To prove the third property, we write

$$P_2(A_\epsilon^{(n)}(P_1||P_2)) = \sum_{x^n \in A_\epsilon^{(n)}(P_1||P_2)} P_2(x_1, x_2, \ldots, x_n) \quad (11.208)$$

$$\leq \sum_{x^n \in A_\epsilon^{(n)}(P_1||P_2)} P_1(x_1, x_2, \ldots, x_n) 2^{-n(D(P_1||P_2)-\epsilon)} \quad (11.209)$$

$$= 2^{-n(D(P_1||P_2)-\epsilon)} \sum_{x^n \in A_\epsilon^{(n)}(P_1||P_2)} P_1(x_1, x_2, \ldots, x_n) \quad (11.210)$$

$$= 2^{-n(D(P_1||P_2)-\epsilon)} P_1(A_\epsilon^{(n)}(P_1||P_2)) \quad (11.211)$$

$$\leq 2^{-n(D(P_1||P_2)-\epsilon)}, \quad (11.212)$$

where the first inequality follows from property 1, and the second inequality follows from the fact that the probability of any set under P_1 is less than 1.

To prove the lower bound on the probability of the relative entropy typical set, we use a parallel argument with a lower bound on the probability:

$$P_2(A_\epsilon^{(n)}(P_1||P_2)) = \sum_{x^n \in A_\epsilon^{(n)}(P_1||P_2)} P_2(x_1, x_2, \ldots, x_n) \quad (11.213)$$

$$\geq \sum_{x^n \in A_\epsilon^{(n)}(P_1||P_2)} P_1(x_1, x_2, \ldots, x_n) 2^{-n(D(P_1||P_2)+\epsilon)} \quad (11.214)$$

$$= 2^{-n(D(P_1||P_2)+\epsilon)} \sum_{x^n \in A_\epsilon^{(n)}(P_1||P_2)} P_1(x_1, x_2, \ldots, x_n) \quad (11.215)$$

$$= 2^{-n(D(P_1||P_2)+\epsilon)} P_1(A_\epsilon^{(n)}(P_1||P_2)) \quad (11.216)$$

$$\geq (1-\epsilon) 2^{-n(D(P_1||P_2)+\epsilon)}, \quad (11.217)$$

where the second inequality follows from the second property of $A_\epsilon^{(n)}(P_1||P_2)$. □

With the standard AEP in Chapter 3, we also showed that any set that has a high probability has a high intersection with the typical set, and therefore has about 2^{nH} elements. We now prove the corresponding result for relative entropy.

Lemma 11.8.1 *Let $B_n \subset \mathcal{X}^n$ be any set of sequences $x_1, x_2, \ldots, x_n$ such that $P_1(B_n) > 1 - \epsilon$. Let P_2 be any other distribution such that $D(P_1||P_2) < \infty$. Then $P_2(B_n) > (1 - 2\epsilon)2^{-n(D(P_1||P_2)+\epsilon)}$.*

Proof: For simplicity, we will denote $A_\epsilon^{(n)}(P_1||P_2)$ by A_n. Since $P_1(B_n) > 1 - \epsilon$ and $P(A_n) > 1 - \epsilon$ (Theorem 11.8.2), we have, by the union of events bound, $P_1(A_n^c \cup B_n^c) < 2\epsilon$, or equivalently, $P_1(A_n \cap B_n) > 1 - 2\epsilon$. Thus,

$$P_2(B_n) \geq P_2(A_n \cap B_n) \tag{11.218}$$

$$= \sum_{x^n \in A_n \cap B_n} P_2(x^n) \tag{11.219}$$

$$\geq \sum_{x^n \in A_n \cap B_n} P_1(x^n) 2^{-n(D(P_1||P_2)+\epsilon)} \tag{11.220}$$

$$= 2^{-n(D(P_1||P_2)+\epsilon)} \sum_{x^n \in A_n \cap B_n} P_1(x^n) \tag{11.221}$$

$$= 2^{-n(D(P_1||P_2)+\epsilon)} P_1(A_n \cap B_n) \tag{11.222}$$

$$\geq 2^{-n(D(P_1||P_2)+\epsilon)}(1 - 2\epsilon), \tag{11.223}$$

where the second inequality follows from the properties of the relative entropy typical sequences (Theorem 11.8.2) and the last inequality follows from the union bound above. □

We now consider the problem of testing two hypotheses, P_1 vs. P_2. We hold one of the probabilities of error fixed and attempt to minimize the other probability of error. We show that the relative entropy is the best exponent in probability of error.

Theorem 11.8.3 (*Chernoff–Stein Lemma*) *Let $X_1, X_2, \ldots, X_n$ be i.i.d. $\sim Q$. Consider the hypothesis test between two alternatives, $Q = P_1$ and $Q = P_2$, where $D(P_1||P_2) < \infty$. Let $A_n \subseteq \mathcal{X}^n$ be an acceptance region for hypothesis H_1. Let the probabilities of error be*

$$\alpha_n = P_1^n(A_n^c), \qquad \beta_n = P_2^n(A_n). \tag{11.224}$$

and for $0 < \epsilon < \frac{1}{2}$, define

$$\beta_n^\epsilon = \min_{\substack{A_n \subseteq \mathcal{X}^n \\ \alpha_n < \epsilon}} \beta_n. \tag{11.225}$$

Then

$$\lim_{n\to\infty} \frac{1}{n} \log \beta_n^{\epsilon} = -D(P_1||P_2). \tag{11.226}$$

Proof: We prove this theorem in two parts. In the first part we exhibit a sequence of sets A_n for which the probability of error β_n goes exponentially to zero as $D(P_1||P_2)$. In the second part we show that no other sequence of sets can have a lower exponent in the probability of error.

For the first part, we choose as the sets $A_n = A_{\epsilon}^{(n)}(P_1||P_2)$. As proved in Theorem 11.8.2, this sequence of sets has $P_1(A_n^c) < \epsilon$ for n large enough. Also,

$$\lim_{n\to\infty} \frac{1}{n} \log P_2(A_n) \leq -(D(P_1||P_2) - \epsilon) \tag{11.227}$$

from property 3 of Theorem 11.8.2. Thus, the relative entropy typical set satisfies the bounds of the lemma.

To show that no other sequence of sets can to better, consider any sequence of sets B_n with $P_1(B_n) > 1 - \epsilon$. By Lemma 11.8.1, we have $P_2(B_n) > (1 - 2\epsilon)2^{-n(D(P_1||P_2)+\epsilon)}$, and therefore

$$\begin{aligned} \lim_{n\to\infty} \frac{1}{n} \log P_2(B_n) &> -(D(P_1||P_2) + \epsilon) + \lim_{n\to\infty} \frac{1}{n} \log(1 - 2\epsilon) \\ &= -(D(P_1||P_2) + \epsilon). \end{aligned} \tag{11.228}$$

Thus, no other sequence of sets has a probability of error exponent better than $D(P_1||P_2)$. Thus, the set sequence $A_n = A_{\epsilon}^{(n)}(P_1||P_2)$ is asymptotically optimal in terms of the exponent in the probability. □

Not that the relative entropy typical set, although asymptotically optimal (i.e., achieving the best asymptotic rate), is not the optimal set for any fixed hypothesis-testing problem. The optimal set that minimizes the probabilities of error is that given by the Neyman–Pearson lemma.

11.9 CHERNOFF INFORMATION

We have considered the problem of hypothesis testing in the classical setting, in which we treat the two probabilities of error separately. In the derivation of the Chernoff–Stein lemma, we set $\alpha_n \leq \epsilon$ and achieved $\beta_n \doteq 2^{-nD}$. But this approach lacks symmetry. Instead, we can follow a Bayesian approach, in which we assign prior probabilities to both

hypotheses. In this case we wish to minimize the overall probability of error given by the weighted sum of the individual probabilities of error. The resulting error exponent is the *Chernoff information*.

The setup is as follows: $X_1, X_2, \ldots, X_n$ i.i.d. $\sim Q$. We have two hypotheses: $Q = P_1$ with prior probability π_1 and $Q = P_2$ with prior probability π_2. The overall probability of error is

$$P_e^{(n)} = \pi_1 \alpha_n + \pi_2 \beta_n. \tag{11.229}$$

Let

$$D^* = \lim_{n \to \infty} -\frac{1}{n} \log \min_{A_n \subseteq \mathcal{X}^n} P_e^{(n)}. \tag{11.230}$$

Theorem 11.9.1 (*Chernoff*) *The best achievable exponent in the Bayesian probability of error is D^*, where*

$$D^* = D(P_{\lambda^*} || P_1) = D(P_{\lambda^*} || P_2), \tag{11.231}$$

with

$$P_\lambda = \frac{P_1^\lambda(x) P_2^{1-\lambda}(x)}{\sum_{a \in \mathcal{X}} P_1^\lambda(a) P_2^{1-\lambda}(a)}, \tag{11.232}$$

and λ^* the value of λ such that

$$D(P_{\lambda^*} || P_1) = D(P_{\lambda^*} || P_2). \tag{11.233}$$

Proof: The basic details of the proof were given in Section 11.8. We have shown that the optimum test is a likelihood ratio test, which can be considered to be of the form

$$D(P_{X^n} || P_2) - D(P_{X^n} || P_1) > \frac{1}{n} \log T. \tag{11.234}$$

The test divides the probability simplex into regions corresponding to hypothesis 1 and hypothesis 2, respectively. This is illustrated in Figure 11.10.

Let A be the set of types associated with hypothesis 1. From the discussion preceding (11.200), it follows that the closest point in the set A^c to P_1 is on the boundary of A and is of the form given by (11.232). Then from the discussion in Section 11.8, it is clear that P_λ is the distribution

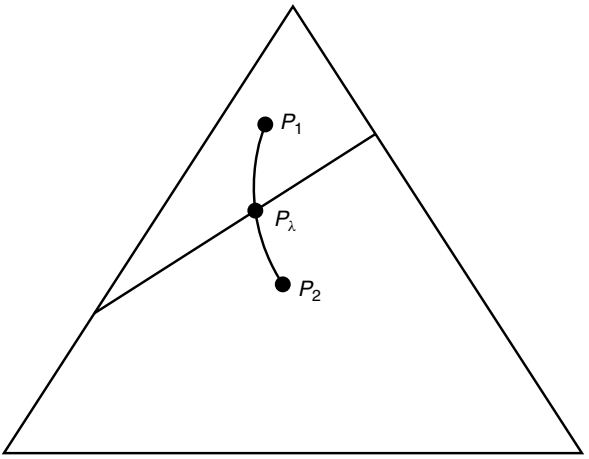

FIGURE 11.10. Probability simplex and Chernoff information.

in A that is closest to P_2; it is also the distribution in A^c that is closest to P_1. By Sanov's theorem, we can calculate the associated probabilities of error,

$$\alpha_n = P_1^n(A^c) \doteq 2^{-nD(P_{\lambda^*}||P_1)} \tag{11.235}$$

and

$$\beta_n = P_2^n(A) \doteq 2^{-nD(P_{\lambda^*}||P_2)}. \tag{11.236}$$

In the Bayesian case, the overall probability of error is the weighted sum of the two probabilities of error,

$$P_e \doteq \pi_1 2^{-nD(P_\lambda||P_1)} + \pi_2 2^{-nD(P_\lambda||P_2)} \doteq 2^{-n\min\{D(P_\lambda||P_1), D(P_\lambda||P_2)\}}, \tag{11.237}$$

since the exponential rate is determined by the worst exponent. Since $D(P_\lambda||P_1)$ increases with λ and $D(P_\lambda||P_2)$ decreases with λ, the maximum value of the minimum of $\{D(P_\lambda||P_1), D(P_\lambda||P_2)\}$ is attained when they are equal. This is illustrated in Figure 11.11. Hence, we choose λ so that

$$D(P_\lambda||P_1) = D(P_\lambda||P_2). \tag{11.238}$$

Thus, $C(P_1, P_2)$ is the highest achievable exponent for the probability of error and is called the Chernoff information. □

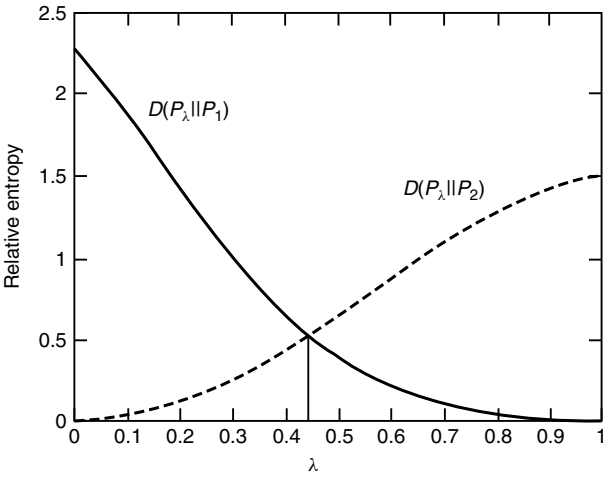

FIGURE 11.11. Relative entropy $D(P_\lambda||P_1)$ and $D(P_\lambda||P_2)$ as a function of λ.

The definition $D^* = D(P_{\lambda *}||P_1) = D(P_{\lambda *}||P_2)$ is equivalent to the standard definition of *Chernoff information*,

$$C(P_1, P_2) \stackrel{\triangle}{=} -\min_{0\le\lambda\le 1} \log\left(\sum_x P_1^\lambda(x) P_2^{1-\lambda}(x)\right). \tag{11.239}$$

It is left as an exercise to the reader to show the equivalence of (11.231) and (11.239).

We outline briefly the usual derivation of the Chernoff information bound. The maximum a posteriori probability decision rule minimizes the Bayesian probability of error. The decision region A for hypothesis H_1 for the maximum a posteriori rule is

$$A = \left\{\mathbf{x} : \frac{\pi_1 P_1(\mathbf{x})}{\pi_2 P_2(\mathbf{x})} > 1\right\}, \tag{11.240}$$

the set of outcomes where the a posteriori probability of hypothesis H_1 is greater than the a posteriori probability of hypothesis H_2. The probability of error for this rule is

$$P_e = \pi_1\alpha_n + \pi_2\beta_n \tag{11.241}$$

$$= \sum_{A^c} \pi_1 P_1 + \sum_A \pi_2 P_2 \tag{11.242}$$

$$= \sum \min\{\pi_1 P_1, \pi_2 P_2\}. \tag{11.243}$$

Now for any two positive numbers a and b, we have

$$\min\{a, b\} \le a^\lambda b^{1-\lambda} \quad \text{for all } 0 \le \lambda \le 1. \tag{11.244}$$

Using this to continue the chain, we have

$$P_e = \sum \min\{\pi_1 P_1, \pi_2 P_2\} \tag{11.245}$$

$$\le \sum (\pi_1 P_1)^\lambda (\pi_2 P_2)^{1-\lambda} \tag{11.246}$$

$$\le \sum P_1^\lambda P_2^{1-\lambda}. \tag{11.247}$$

For a sequence of i.i.d. observations, $P_k(\mathbf{x}) = \prod_{i=1}^n P_k(x_i)$, and

$$P_e^{(n)} \le \sum \pi_1^\lambda \pi_2^{1-\lambda} \prod_i P_1^\lambda(x_i) P_2^{1-\lambda}(x_i) \tag{11.248}$$

$$= \pi_1^\lambda \pi_2^{1-\lambda} \prod_i \sum P_1^\lambda(x_i) P_2^{1-\lambda}(x_i) \tag{11.249}$$

$$\le \prod_{x_i} \sum P_1^\lambda P_2^{1-\lambda} \tag{11.250}$$

$$= \left(\sum_x P_1^\lambda P_2^{1-\lambda} \right)^n, \tag{11.251}$$

where (11.250) follows since $\pi_1 \le 1, \pi_2 \le 1$. Hence, we have

$$\frac{1}{n} \log P_e^{(n)} \le \log \sum P_1^\lambda(x) P_2^{1-\lambda}(x). \tag{11.252}$$

Since this is true for all λ, we can take the minimum over $0 \le \lambda \le 1$, resulting in the Chernoff information bound. This proves that the exponent is no better than $C(P_1, P_2)$. Achievability follows from Theorem 11.9.1.

Note that the Bayesian error exponent does not depend on the actual value of π_1 and π_2, as long as they are nonzero. Essentially, the effect of the prior is washed out for large sample sizes. The optimum decision rule is to choose the hypothesis with the maximum a posteriori probability, which corresponds to the test

$$\frac{\pi_1 P_1(X_1, X_2, \ldots, X_n)}{\pi_2 P_2(X_1, X_2, \ldots, X_n)} \gtrless 1. \tag{11.253}$$

Taking the log and dividing by n, this test can be rewritten as

$$\frac{1}{n}\log\frac{\pi_1}{\pi_2}+\frac{1}{n}\sum_i \log\frac{P_1(X_i)}{P_2(X_i)}\lessgtr 0, \tag{11.254}$$

where the second term tends to $D(P_1||P_2)$ or $-D(P_2||P_1)$ accordingly as P_1 or P_2 is the true distribution. The first term tends to 0, and the effect of the prior distribution washes out.

Finally, to round off our discussion of large deviation theory and hypothesis testing, we consider an example of the conditional limit theorem.

Example 11.9.1 Suppose that major league baseball players have a batting average of 260 with a standard deviation of 15 and suppose that minor league ballplayers have a batting average of 240 with a standard deviation of 15. A group of 100 ballplayers from one of the leagues (the league is chosen at random) are found to have a group batting average greater than 250 and are therefore judged to be major leaguers. We are now told that we are mistaken; these players are minor leaguers. What can we say about the distribution of batting averages among these 100 players? The conditional limit theorem can be used to show that the distribution of batting averages among these players will have a mean of 250 and a standard deviation of 15. To see this, we abstract the problem as follows.

Let us consider an example of testing between two Gaussian distributions, $f_1 = \mathcal{N}(1, \sigma^2)$ and $f_2 = \mathcal{N}(-1, \sigma^2)$, with different means and the same variance. As discussed in Section 11.8, the likelihood ratio test in this case is equivalent to comparing the sample mean with a threshold. The Bayes test is "Accept the hypothesis $f = f_1$ if $\frac{1}{n}\sum_{i=1}^n X_i > 0$." Now assume that we make an error of the first kind (we say that $f = f_1$ when indeed $f = f_2$) in this test. What is the conditional distribution of the samples given that we have made an error?

We might guess at various possibilities:

- The sample will look like a $(\frac{1}{2}, \frac{1}{2})$ mix of the two normal distributions. Plausible as this is, it is incorrect.
- $X_i \approx 0$ for all i. This is quite clearly very unlikely, although it is conditionally likely that $\overline{X}_n$ is close to 0.
- The correct answer is given by the conditional limit theorem. If the true distribution is f_2 and the sample type is in the set A, the conditional distribution is close to f^*, the distribution in A that is closest to f_2. By symmetry, this corresponds to $\lambda = \frac{1}{2}$ in (11.232). Calculating

the distribution, we get

$$f^*(x) = \frac{\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-1)^2}{2\sigma^2}}\right)^{\frac{1}{2}} \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x+1)^2}{2\sigma^2}}\right)^{\frac{1}{2}}}{\int \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-1)^2}{2\sigma^2}}\right)^{\frac{1}{2}} \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x+1)^2}{2\sigma^2}}\right)^{\frac{1}{2}} dx} \tag{11.255}$$

$$= \frac{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x^2+1)}{2\sigma^2}}}{\int \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x^2+1)}{2\sigma^2}} dx} \tag{11.256}$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \tag{11.257}$$

$$= \mathcal{N}(0, \sigma^2). \tag{11.258}$$

It is interesting to note that the conditional distribution is normal with mean 0 and with the same variance as the original distributions. This is strange but true; if we mistake a normal population for another, the "shape" of this population still looks normal with the same variance and a different mean. Apparently, this rare event does not result from bizarre-looking data.

Example 11.9.2 (*Large deviation theory and football*) Consider a very simple version of football in which the score is directly related to the number of yards gained. Assume that the coach has a choice between two strategies: running or passing. Associated with each strategy is a distribution on the number of yards gained. For example, in general, running

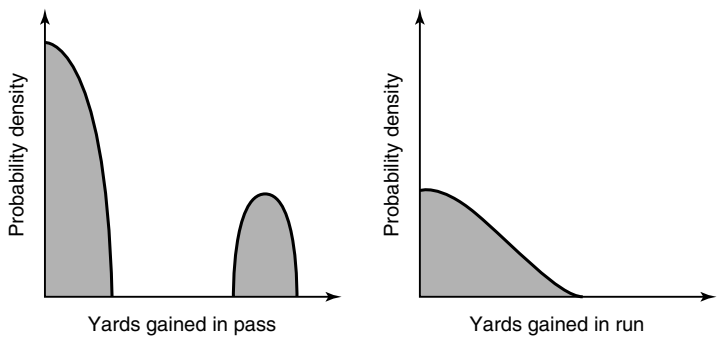

FIGURE 11.12. Distribution of yards gained in a run or a pass play.

results in a gain of a few yards with very high probability, whereas passing results in huge gains with low probability. Examples of the distributions are illustrated in Figure 11.12.

At the beginning of the game, the coach uses the strategy that promises the greatest expected gain. Now assume that we are in the closing minutes of the game and one of the teams is leading by a large margin. (Let us ignore first downs and adaptable defenses.) So the trailing team will win only if it is very lucky. If luck is required to win, we might as well assume that we will be lucky and play accordingly. What is the appropriate strategy?

Assume that the team has only n plays left and it must gain l yards, where l is much larger than n times the expected gain under each play. The probability that the team succeeds in achieving l yards is exponentially small; hence, we can use the large deviation results and Sanov's theorem to calculate the probability of this event. To be precise, we wish to calculate the probability that $\sum_{i=1}^{n} Z_i \geq n\alpha$, where Z_i are independent random variables and Z_i has a distribution corresponding to the strategy chosen.

The situation is illustrated in Figure 11.13. Let E be the set of types corresponding to the constraint,

$$E = \left\{ P : \sum_{a \in \mathcal{X}} P(a)a \geq \alpha \right\}. \tag{11.259}$$

If P_1 is the distribution corresponding to passing all the time, the probability of winning is the probability that the sample type is in E, which by Sanov's theorem is $2^{-nD(P_1^*||P_1)}$, where P_1^* is the distribution in E that is closest to P_1. Similarly, if the coach uses the running game all the time,

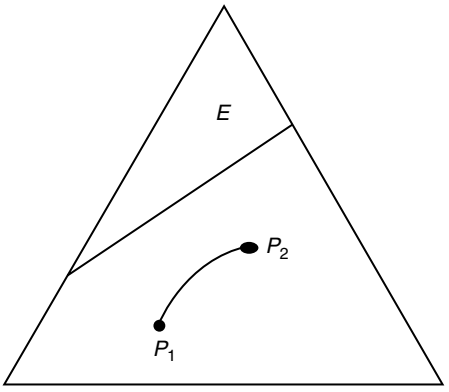

FIGURE 11.13. Probability simplex for a football game.

the probability of winning is $2^{-nD(P_2^*||P_2)}$. What if he uses a mixture of strategies? Is it possible that $2^{-nD(P_\lambda^*||P_\lambda)}$, the probability of winning with a mixed strategy, $P_\lambda = \lambda P_1 + (1-\lambda)P_2$, is better than the probability of winning with either pure passing or pure running? The somewhat surprising answer is yes, as can be shown by example. This provides a reason to use a mixed strategy other than the fact that it confuses the defense.

We end this section with another inequality due to Chernoff, which is a special version of Markov's inequality. This inequality is called the *Chernoff bound*.

Lemma 11.9.1 *Let Y be any random variable and let $\psi(s)$ be the moment generating function of Y,*

$$\psi(s) = Ee^{sY}. \tag{11.260}$$

Then for all $s \geq 0$,

$$\Pr(Y \geq a) \leq e^{-sa}\psi(s), \tag{11.261}$$

and thus

$$\Pr(Y \geq a) \leq \min_{s \geq 0} e^{-sa}\psi(s). \tag{11.262}$$

Proof: Apply Markov's inequality to the nonnegative random variable e^{sY}. □

11.10 FISHER INFORMATION AND THE CRAMÉR–RAO INEQUALITY

A standard problem in statistical estimation is to determine the parameters of a distribution from a sample of data drawn from that distribution. For example, let $X_1, X_2, \ldots, X_n$ be drawn i.i.d. $\sim \mathcal{N}(\theta, 1)$. Suppose that we wish to estimate θ from a sample of size n. There are a number of functions of the data that we can use to estimate θ. For example, we can use the first sample X_1. Although the expected value of X_1 is θ, it is clear that we can do better by using more of the data. We guess that the best estimate of θ is the sample mean $\overline{X}_n = \frac{1}{n}\sum X_i$. Indeed, it can be shown that $\overline{X}_n$ is the minimum mean-squared-error unbiased estimator.

We begin with a few definitions. Let $\{f(x;\theta)\}$, $\theta \in \Theta$, denote an indexed family of densities, $f(x;\theta) \geq 0$, $\int f(x;\theta)\,dx = 1$ for all $\theta \in \Theta$. Here Θ is called the *parameter set*.

Definition An *estimator* for θ for sample size n is a function $T : \mathcal{X}^n \to \Theta$.

An estimator is meant to approximate the value of the parameter. It is therefore desirable to have some idea of the goodness of the approximation. We will call the difference $T - \theta$ the *error* of the estimator. The error is a random variable.

Definition The *bias* of an estimator $T(X_1, X_2, \ldots, X_n)$ for the parameter θ is the expected value of the error of the estimator [i.e., the bias is $E_\theta T(x_1, x_2, \ldots, x_n) - \theta$]. The subscript θ means that the expectation is with respect to the density $f(\cdot; \theta)$. The estimator is said to be *unbiased* if the bias is zero for all $\theta \in \Theta$ (i.e., the expected value of the estimator is equal to the parameter).

Example 11.10.1 Let $X_1, X_2, \ldots, X_n$ drawn i.i.d. $\sim f(x) = (1/\lambda) e^{-x/\lambda}$, $x \geq 0$ be a sequence of exponentially distributed random variables. Estimators of λ include X_1 and $\overline{X}_n$. Both estimators are unbiased.

The bias is the expected value of the error, and the fact that it is zero does not guarantee that the error is low with high probability. We need to look at some loss function of the error; the most commonly chosen loss function is the expected square of the error. A good estimator should have a low expected squared error and should have an error that approaches 0 as the sample size goes to infinity. This motivates the following definition:

Definition An estimator $T(X_1, X_2, \ldots, X_n)$ for θ is said to be *consistent in probability* if
$T(X_1, X_2, \ldots, X_n) \to \theta$ in probability as $n \to \infty$.

Consistency is a desirable asymptotic property, but we are interested in the behavior for small sample sizes as well. We can then rank estimators on the basis of their mean-squared error.

Definition An estimator $T_1(X_1, X_2, \ldots, X_n)$ is said to *dominate* another estimator $T_2(X_1, X_2, \ldots, X_n)$ if, for all θ,

$$E\,(T_1(X_1, X_2, \ldots, X_n) - \theta)^2 \leq E\,(T_2(X_1, X_2, \ldots, X_n) - \theta)^2\,. \tag{11.263}$$

This raises a natural question: Is there a best estimator of θ that dominates every other estimator? To answer this question, we derive the Cramér–Rao lower bound on the mean-squared error of any estimator. We first define the score function of the distribution $f(x; \theta)$. We then use the Cauchy–Schwarz inequality to prove the Cramér–Rao lower bound on the variance of all unbiased estimators.

Definition The *score* V is a random variable defined by

$$V = \frac{\partial}{\partial \theta} \ln f(X;\theta) = \frac{\frac{\partial}{\partial \theta} f(X;\theta)}{f(X;\theta)}, \tag{11.264}$$

where $X \sim f(x;\theta)$.

The mean value of the score is

$$EV = \int \frac{\frac{\partial}{\partial \theta} f(x;\theta)}{f(x;\theta)} f(x;\theta)\, dx \tag{11.265}$$

$$= \int \frac{\partial}{\partial \theta} f(x;\theta)\, dx \tag{11.266}$$

$$= \frac{\partial}{\partial \theta} \int f(x;\theta)\, dx \tag{11.267}$$

$$= \frac{\partial}{\partial \theta} 1 \tag{11.268}$$

$$= 0, \tag{11.269}$$

and therefore $EV^2 = \text{var}(V)$. The variance of the score has a special significance.

Definition The *Fisher information* $J(\theta)$ is the variance of the score:

$$J(\theta) = E_\theta \left[\frac{\partial}{\partial \theta} \ln f(X;\theta) \right]^2. \tag{11.270}$$

If we consider a sample of n random variables $X_1, X_2, \ldots, X_n$ drawn i.i.d. $\sim f(x;\theta)$, we have

$$f(x_1, x_2, \ldots, x_n; \theta) = \prod_{i=1}^{n} f(x_i;\theta), \tag{11.271}$$

and the score function is the sum of the individual score functions,

$$V(X_1, X_2, \ldots, X_n) = \frac{\partial}{\partial \theta} \ln f(X_1, X_2, \ldots, X_n; \theta) \tag{11.272}$$

$$= \sum_{i=1}^{n} \frac{\partial}{\partial \theta} \ln f(X_i;\theta) \tag{11.273}$$

$$= \sum_{i=1}^{n} V(X_i), \tag{11.274}$$

where the $V(X_i)$ are independent, identically distributed with zero mean. Hence, the n-sample Fisher information is

$$J_n(\theta) = E_\theta\left[\frac{\partial}{\partial\theta}\ln f(X_1, X_2, \ldots, X_n; \theta)\right]^2 \tag{11.275}$$

$$= E_\theta V^2(X_1, X_2, \ldots, X_n) \tag{11.276}$$

$$= E_\theta\left(\sum_{i=1}^{n} V(X_i)\right)^2 \tag{11.277}$$

$$= \sum_{i=1}^{n} E_\theta V^2(X_i) \tag{11.278}$$

$$= nJ(\theta). \tag{11.279}$$

Consequently, the Fisher information for n i.i.d. samples is n times the individual Fisher information. The significance of the Fisher information is shown in the following theorem.

Theorem 11.10.1 (*Cramér–Rao inequality*) *The mean-squared error of any unbiased estimator $T(X)$ of the parameter θ is lower bounded by the reciprocal of the Fisher information:*

$$\text{var}(T) \geq \frac{1}{J(\theta)}. \tag{11.280}$$

Proof: Let V be the score function and T be the estimator. By the Cauchy–Schwarz inequality, we have

$$(E_\theta[(V - E_\theta V)(T - E_\theta T)])^2 \leq E_\theta(V - E_\theta V)^2 E_\theta(T - E_\theta T)^2. \tag{11.281}$$

Since T is unbiased, $E_\theta T = \theta$ for all θ. By (11.269), $E_\theta V = 0$ and hence $E_\theta(V - E_\theta V)(T - E_\theta T) = E_\theta(VT)$. Also, by definition, $\text{var}(V) = J(\theta)$. Substituting these conditions in (11.281), we have

$$[E_\theta(VT)]^2 \leq J(\theta)\text{var}(T). \tag{11.282}$$

Now,

$$E_\theta(VT) = \int \frac{\frac{\partial}{\partial\theta}f(x;\theta)}{f(x;\theta)} T(x) f(x;\theta)\, dx \tag{11.283}$$

$$= \int \frac{\partial}{\partial \theta} f(x;\theta) T(x)\, dx \tag{11.284}$$

$$= \frac{\partial}{\partial \theta} \int f(x;\theta) T(x)\, dx \tag{11.285}$$

$$= \frac{\partial}{\partial \theta} E_\theta T \tag{11.286}$$

$$= \frac{\partial}{\partial \theta} \theta \tag{11.287}$$

$$= 1, \tag{11.288}$$

where the interchange of differentiation and integration in (11.285) can be justified using the bounded convergence theorem for appropriately well behaved $f(x;\theta)$, and (11.287) follows from the fact that the estimator T is unbiased. Substituting this in (11.282), we obtain

$$\text{var}(T) \geq \frac{1}{J(\theta)}, \tag{11.289}$$

which is the Cramér–Rao inequality for unbiased estimators. □

By essentially the same arguments, we can show that for any estimator

$$E(T-\theta)^2 \geq \frac{(1+b_T'(\theta))^2}{J(\theta)} + b_T^2(\theta), \tag{11.290}$$

where $b_T(\theta) = E_\theta T - \theta$ and $b_T'(\theta)$ is the derivative of $b_T(\theta)$ with respect to θ. The proof of this is left as a problem at the end of the chapter.

Example 11.10.2 Let $X_1, X_2, \ldots, X_n$ be i.i.d. $\sim \mathcal{N}(\theta, \sigma^2)$, σ^2 known. Here $J(\theta) = n/\sigma^2$. Let $T(X_1, X_2, \ldots, X_n) = \overline{X}_n = \frac{1}{n}\sum X_i$. Then $E_\theta(\overline{X}_n - \theta)^2 = \sigma^2/n = 1/J(\theta)$. Thus, $\overline{X}_n$ is the minimum variance unbiased estimator of θ, since it achieves the Cramér–Rao lower bound.

The Cramér–Rao inequality gives us a lower bound on the variance for all unbiased estimators. When this bound is achieved, we call the estimator efficient.

Definition An unbiased estimator T is said to be *efficient* if it meets the Cramér–Rao bound with equality [i.e., if $\text{var}(T) = \frac{1}{J(\theta)}$].

The Fisher information is therefore a measure of the amount of "information" about θ that is present in the data. It gives a lower bound on the error in estimating θ from the data. However, it is possible that there does not exist an estimator meeting this lower bound.

We can generalize the concept of Fisher information to the multiparameter case, in which case we define the Fisher information matrix $J(\theta)$ with elements

$$J_{ij}(\theta) = \int f(x;\theta)\frac{\partial}{\partial\theta_i}\ln f(x;\theta)\frac{\partial}{\partial\theta_j}\ln f(x;\theta)\ dx. \tag{11.291}$$

The Cramér–Rao inequality becomes the matrix inequality

$$\Sigma \geq J^{-1}(\theta), \tag{11.292}$$

where Σ is the covariance matrix of a set of unbiased estimators for the parameters θ and $\Sigma \geq J^{-1}(\theta)$ in the sense that the difference $\Sigma - J^{-1}$ is a nonnegative definite matrix. We will not go into the details of the proof for multiple parameters; the basic ideas are similar.

Is there a relationship between the Fisher information $J(\theta)$ and quantities such as entropy defined earlier? Note that Fisher information is defined with respect to a family of parametric distributions, unlike entropy, which is defined for all distributions. But we can parametrize any distribution $f(x)$ by a location parameter θ and define Fisher information with respect to the family of densities $f(x-\theta)$ under translation. We explore the relationship in greater detail in Section 17.8, where we show that while entropy is related to the volume of the typical set, the Fisher information is related to the surface area of the typical set. Further relationships of Fisher information to relative entropy are developed in the problems.

SUMMARY

Basic identities

$$Q^n(\mathbf{x}) = 2^{-n(D(P_{\mathbf{x}}||Q)+H(P_{\mathbf{x}}))}, \tag{11.293}$$

$$|\mathcal{P}_n| \leq (n+1)^{|\mathcal{X}|}, \tag{11.294}$$

$$|T(P)| \doteq 2^{nH(P)}, \tag{11.295}$$

$$Q^n(T(P)) \doteq 2^{-nD(P||Q)}. \tag{11.296}$$

Universal data compression

$$P_e^{(n)} \leq 2^{-nD(P_R^*||Q)} \quad \text{for all } Q, \tag{11.297}$$

where

$$D(P_R^*||Q) = \min_{P:H(P)\geq R} D(P||Q). \tag{11.298}$$

Large deviations (Sanov's theorem)

$$Q^n(E) = Q^n(E \cap \mathcal{P}_n) \leq (n+1)^{|\mathcal{X}|} 2^{-nD(P^*||Q)}, \tag{11.299}$$

$$D(P^*||Q) = \min_{P\in E} D(P||Q). \tag{11.300}$$

If E is the closure of its interior, then

$$Q^n(E) \doteq 2^{-nD(P^*||Q)}. \tag{11.301}$$

$\mathcal{L}_1$ bound on relative entropy

$$D(P_1||P_2) \geq \frac{1}{2\ln 2}||P_1 - P_2||_1^2. \tag{11.302}$$

Pythagorean theorem. If E is a convex set of types, distribution $Q \notin E$, and P^* achieves $D(P^*||Q) = \min_{P\in E} D(P||Q)$, we have

$$D(P||Q) \geq D(P||P^*) + D(P^*||Q) \tag{11.303}$$

for all $P \in E$.

Conditional limit theorem. If $X_1, X_2, \ldots, X_n$ i.i.d. $\sim Q$, then

$$\Pr(X_1 = a | P_{X^n} \in E) \to P^*(a) \qquad \text{in probability,} \tag{11.304}$$

where P^* minimizes $D(P||Q)$ over $P \in E$. In particular,

$$\Pr\left\{ X_1 = a \,\middle|\, \frac{1}{n}\sum_{i=1}^{n} X_i \geq \alpha \right\} \to \frac{Q(a)e^{\lambda a}}{\sum_x Q(x)e^{\lambda x}}. \tag{11.305}$$

Neyman–Pearson lemma. The optimum test between two densities P_1 and P_2 has a decision region of the form "accept $P = P_1$ if $\frac{P_1(x_1,x_2,\ldots,x_n)}{P_2(x_1,x_2,\ldots,x_n)} > T$."

Chernoff–Stein lemma. The best achievable error exponent β_n^ϵ if $\alpha_n \leq \epsilon$:

$$\beta_n^\epsilon = \min_{\substack{A_n \subseteq \mathcal{X}^n \\ \alpha_n < \epsilon}} \beta_n, \tag{11.306}$$

$$\lim_{n\to\infty} \frac{1}{n} \log \beta_n^\epsilon = -D(P_1||P_2). \tag{11.307}$$

Chernoff information. The best achievable exponent for a Bayesian probability of error is

$$D^* = D(P_{\lambda^*}||P_1) = D(P_{\lambda^*}||P_2), \tag{11.308}$$

where

$$P_\lambda = \frac{P_1^\lambda(x) P_2^{1-\lambda}(x)}{\sum_{a\in\mathcal{X}} P_1^\lambda(a) P_2^{1-\lambda}(a)} \tag{11.309}$$

with $\lambda = \lambda^*$ chosen so that

$$D(P_\lambda||P_1) = D(P_\lambda||P_2). \tag{11.310}$$

Fisher information

$$J(\theta) = E_\theta\left[\frac{\partial}{\partial\theta} \ln f(x;\theta)\right]^2. \tag{11.311}$$

Cramér–Rao inequality. For any unbiased estimator T of θ,

$$E_\theta(T(X) - \theta)^2 = \text{var}(T) \geq \frac{1}{J(\theta)}. \tag{11.312}$$

PROBLEMS

11.1 *Chernoff–Stein lemma.* Consider the two-hypothesis test

$$H_1 : f = f_1 \quad \text{vs.} \quad H_2 : f = f_2.$$

Find $D(f_1 \parallel f_2)$ if

(a) $f_i(x) = N(0, \sigma_i^2), i = 1, 2.$

(b) $f_i(x) = \lambda_i e^{-\lambda_i x}, x \geq 0, i = 1, 2.$

(c) $f_1(x)$ is the uniform density over the interval $[0, 1]$ and $f_2(x)$ is the uniform density over $[a, a + 1]$. Assume that $0 < a < 1$.

(d) f_1 corresponds to a fair coin and f_2 corresponds to a two-headed coin.

11.2 *Relation between $D(P \parallel Q)$ and chi-square.* Show that the χ^2 statistic

$$\chi^2 = \Sigma_x \frac{(P(x) - Q(x))^2}{Q(x)}$$

is (twice) the first term in the Taylor series expansion of $D(P \parallel Q)$ about Q. Thus, $D(P \parallel Q) = \frac{1}{2}\chi^2 + \cdots$. [*Suggestion:* Write $\frac{P}{Q} = 1 + \frac{P-Q}{Q}$ and expand the log.]

11.3 *Error exponent for universal codes.* A universal source code of rate R achieves a probability of error $P_e^{(n)} \doteq e^{-nD(P^* \| Q)}$, where Q is the true distribution and P^* achieves $\min D(P \parallel Q)$ over all P such that $H(P) \geq R$.

(a) Find P^* in terms of Q and R.

(b) Now let X be binary. Find the region of source probabilities $Q(x)$, $x \in \{0, 1\}$, for which rate R is sufficient for the universal source code to achieve $P_e^{(n)} \to 0$.

11.4 *Sequential projection.* We wish to show that projecting Q onto P_1 and then projecting the projection $\hat{Q}$ onto $P_1 \bigcap P_2$ is the same as projecting Q directly onto $P_1 \bigcap P_2$. Let $\mathcal{P}_1$ be the set of probability mass functions on $\mathcal{X}$ satisfying

$$\sum_x p(x) = 1, \tag{11.313}$$

$$\sum_x p(x) h_i(x) \geq \alpha_i, \qquad i = 1, 2, \ldots, r. \tag{11.314}$$

Let $\mathcal{P}_2$ be the set of probability mass functions on $\mathcal{X}$ satisfying

$$\sum_x p(x) = 1, \tag{11.315}$$

$$\sum_x p(x) g_j(x) \geq \beta_j, \qquad j = 1, 2, \ldots, s. \tag{11.316}$$

Suppose that $Q \notin P_1 \bigcup P_2$. Let P^* minimize $D(P \parallel Q)$ over all $P \in \mathcal{P}_1$. Let R^* minimize $D(R \parallel Q)$ over all $R \in \mathcal{P}_1 \bigcap \mathcal{P}_2$. Argue that R^* minimizes $D(R \parallel P^*)$ over all $R \in P_1 \bigcap P_2$.

11.5 *Counting*. Let $\mathcal{X} = \{1, 2, \ldots, m\}$. Show that the number of sequences $x^n \in \mathcal{X}^n$ satisfying $\frac{1}{n}\sum_{i=1}^{n} g(x_i) \geq \alpha$ is approximately equal to 2^{nH^*}, to first order in the exponent, for n sufficiently large, where

$$H^* = \max_{P:\sum_{i=1}^{m} P(i)g(i) \geq \alpha} H(P). \tag{11.317}$$

11.6 *Biased estimates may be better*. Consider the problem of estimating μ and σ^2 from n samples of data drawn i.i.d. from a $\mathcal{N}(\mu, \sigma^2)$ distribution.

(a) Show that $\overline{X}_n$ is an unbiased estimator of μ.

(b) Show that the estimator

$$S_n^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X}_n)^2 \tag{11.318}$$

is a biased estimator of σ^2 and the estimator

$$S_{n-1}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X}_n)^2 \tag{11.319}$$

is unbiased.

(c) Show that S_n^2 has a lower mean-squared error than that of S_{n-1}^2. This illustrates the idea that a biased estimator may be "better" than an unbiased estimator.

11.7 *Fisher information and relative entropy*. Show for a parametric family $\{p_\theta(x)\}$ that

$$\lim_{\theta' \to \theta} \frac{1}{(\theta - \theta')^2} D(p_\theta || p_{\theta'}) = \frac{1}{\ln 4} J(\theta). \tag{11.320}$$

11.8 *Examples of Fisher information*. The Fisher information $J(\Theta)$ for the family $f_\theta(x), \theta \in \mathbf{R}$ is defined by

$$J(\theta) = E_\theta \left(\frac{\partial f_\theta(X)/\partial\theta}{f_\theta(X)} \right)^2 = \int \frac{(f_\theta')^2}{f_\theta}.$$

Find the Fisher information for the following families:

(a) $f_\theta(x) = N(0, \theta) = \frac{1}{\sqrt{2\pi\theta}} e^{-\frac{x^2}{2\theta}}$

(b) $f_\theta(x) = \theta e^{-\theta x}$, $x \geq 0$

(c) What is the Cramèr–Rao lower bound on $E_\theta(\hat{\theta}(X) - \theta)^2$, where $\hat{\theta}(X)$ is an unbiased estimator of θ for parts (a) and (b)?

11.9 *Two conditionally independent looks double the Fisher information.* Let $g_\theta(x_1, x_2) = f_\theta(x_1) f_\theta(x_2)$. Show that $J_g(\theta) = 2J_f(\theta)$.

11.10 *Joint distributions and product distributions.* Consider a joint distribution $Q(x, y)$ with marginals $Q(x)$ and $Q(y)$. Let E be the set of types that look jointly typical with respect to Q:

$$\begin{aligned} E = \{P(x, y) : &- \sum_{x,y} P(x, y) \log Q(x) - H(X) = 0, \\ &- \sum_{x,y} P(x, y) \log Q(y) - H(Y) = 0, \\ &- \sum_{x,y} P(x, y) \log Q(x, y) \\ &- H(X, Y) = 0\}. \end{aligned} \tag{11.321}$$

(a) Let $Q_0(x, y)$ be another distribution on $\mathcal{X} \times \mathcal{Y}$. Argue that the distribution P^* in E that is closest to Q_0 is of the form

$$P^*(x, y) = Q_0(x, y) e^{\lambda_0 + \lambda_1 \log Q(x) + \lambda_2 \log Q(y) + \lambda_3 \log Q(x,y)}, \tag{11.322}$$

where λ_0, λ_1, λ_2, and λ_3 are chosen to satisfy the constraints. Argue that this distribution is unique.

(b) Now let $Q_0(x, y) = Q(x)Q(y)$. Verify that $Q(x, y)$ is of the form (11.322) and satisfies the constraints. Thus, $P^*(x, y) = Q(x, y)$ (i.e., the distribution in E closest to the product distribution is the joint distribution).

11.11 *Cramér–Rao inequality with a bias term.* Let $X \sim f(x; \theta)$ and let $T(X)$ be an estimator for θ. Let $b_T(\theta) = E_\theta T - \theta$ be the bias of the estimator. Show that

$$E(T - \theta)^2 \geq \frac{[1 + b_T'(\theta)]^2}{J(\theta)} + b_T^2(\theta). \tag{11.323}$$

11.12 *Hypothesis testing*. Let $X_1, X_2, \ldots, X_n$ be i.i.d. $\sim p(x)$. Consider the hypothesis test $H_1 : p = p_1$ vs. $H_2 : p = p_2$. Let

$$p_1(x) = \begin{cases} \frac{1}{2}, & x = -1 \\ \frac{1}{4}, & x = 0 \\ \frac{1}{4}, & x = 1 \end{cases}$$

and

$$p_2(x) = \begin{cases} \frac{1}{4}, & x = -1 \\ \frac{1}{4}, & x = 0 \\ \frac{1}{2}, & x = 1. \end{cases}$$

Find the error exponent for Pr{Decide $H_2|H_1$ true} in the best hypothesis test of H_1 vs. H_2 subject to Pr{Decide $H_1|H_2$ true} $\leq \frac{1}{2}$.

11.13 *Sanov's theorem*. Prove a simple version of Sanov's theorem for Bernoulli(q) random variables.
Let the proportion of 1's in the sequence $X_1, X_2, \ldots, X_n$ be

$$\overline{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i. \tag{11.324}$$

By the law of large numbers, we would expect $\overline{X}_n$ to be close to q for large n. Sanov's theorem deals with the probability that p_{X^n} is far away from q. In particular, for concreteness, if we take $p > q > \frac{1}{2}$, Sanov's theorem states that

$$\begin{aligned} & -\frac{1}{n} \log \Pr\left\{(X_1, X_2, \ldots, X_n) : \overline{X}_n \geq p\right\} \\ & \rightarrow p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q} \\ & = D((p, 1-p)||(q, 1-q)). \end{aligned} \tag{11.325}$$

Justify the following steps:

- $$\Pr\left\{(X_1, X_2, \ldots, X_n) : \overline{X}_n \geq p\right\} \leq \sum_{i=\lfloor np \rfloor}^{n} \binom{n}{i} q^i (1-q)^{n-i}. \tag{11.326}$$

- Argue that the term corresponding to $i = \lfloor np \rfloor$ is the largest term in the sum on the right-hand side of the last equation.
- Show that this term is approximately 2^{-nD}.
- Prove an upper bound on the probability in Sanov's theorem using the steps above. Use similar arguments to prove a lower bound and complete the proof of Sanov's theorem.

11.14 *Sanov*. Let X_i be i.i.d. $\sim N(0, \sigma^2)$.

(a) Find the exponent in the behavior of $\Pr\{\frac{1}{n}\sum_{i=1}^{n} X_i^2 \geq \alpha^2\}$. This can be done from first principles (since the normal distribution is nice) or by using Sanov's theorem.

(b) What do the data look like if $\frac{1}{n}\sum_{i=1}^{n} X_i^2 \geq \alpha$? That is, what is the P^* that minimizes $D(P \parallel Q)$?

11.15 *Counting states*. Suppose that an atom is equally likely to be in each of six states, $X \in \{s_1, s_2, s_3, \ldots, s_6\}$. One observes n atoms $X_1, X_2, \ldots, X_n$ independently drawn according to this uniform distribution. It is observed that the frequency of occurrence of state s_1 is twice the frequency of occurrence of state s_2.

(a) To first order in the exponent, what is the probability of observing this event?

(b) Assuming n large, find the conditional distribution of the state of the first atom X_1, given this observation.

11.16 *Hypothesis testing*. Let $\{X_i\}$ be i.i.d. $\sim p(x)$, $x \in \{1, 2, \ldots\}$. Consider two hypotheses, $H_0 : p(x) = p_0(x)$ vs. $H_1 : p(x) = p_1(x)$, where $p_0(x) = \left(\frac{1}{2}\right)^x$ and $p_1(x) = qp^{x-1}$, $x = 1, 2, 3, \ldots$.

(a) Find $D(p_0 \parallel p_1)$.

(b) Let $\Pr\{H_0\} = \frac{1}{2}$. Find the minimal probability of error test for H_0 vs. H_1 given data $X_1, X_2, \ldots, X_n \sim p(x)$.

11.17 *Maximum likelihood estimation*. Let $\{f_\theta(x)\}$ denote a parametric family of densities with parameter $\theta \in \mathcal{R}$. Let $X_1, X_2, \ldots, X_n$ be i.i.d. $\sim f_\theta(x)$. The function

$$l_\theta(x^n) = \ln\left(\prod_{i=1}^{n} f_\theta(x_i)\right)$$

is known as the *log likelihood function*. Let θ_0 denote the true parameter value.

(a) Let the expected log likelihood be

$$E_{\theta_0} l_\theta(X^n) = \int \left(\ln \prod_{i=1}^{n} f_\theta(x_i) \right) \prod_{i=1}^{n} f_{\theta_0}(x_i) dx^n,$$

and show that

$$E_{\theta_0}(l(X^n)) = (-h(f_{\theta_0}) - D(f_{\theta_0} || f_\theta))n.$$

(b) Show that the maximum over θ of the expected log likelihood is achieved by $\theta = \theta_0$.

11.18 *Large deviations*. Let $X_1, X_2, \ldots$ be i.i.d. random variables drawn according to the geometric distribution

$$\Pr\{X = k\} = p^{k-1}(1-p), \qquad k = 1, 2, \ldots.$$

Find good estimates (to first order in the exponent) of:

(a) $\Pr\{\frac{1}{n}\sum_{i=1}^{n} X_i \geq \alpha\}$.

(b) $\Pr\{X_1 = k | \frac{1}{n}\sum_{i=1}^{n} X_i \geq \alpha\}$.

(c) Evaluate parts (a) and (b) for $p = \frac{1}{2}, \alpha = 4$.

11.19 *Another expression for Fisher information*. Use integration by parts to show that

$$J(\theta) = -E \frac{\partial^2 \ln f_\theta(x)}{\partial \theta^2}.$$

11.20 *Stirling's approximation*. Derive a weak form of Stirling's approximation for factorials; that is, show that

$$\left(\frac{n}{e}\right)^n \leq n! \leq n \left(\frac{n}{e}\right)^n \tag{11.327}$$

using the approximation of integrals by sums. Justify the following steps:

$$\ln(n!) = \sum_{i=2}^{n-1} \ln(i) + \ln(n) \leq \int_2^{n-1} \ln x \, dx + \ln n = \cdots \tag{11.328}$$

and

$$\ln(n!) = \sum_{i=1}^{n} \ln(i) \geq \int_0^n \ln x \, dx = \cdots. \tag{11.329}$$

11.21 *Asymptotic value of* $\binom{n}{k}$. Use the simple approximation of Problem 11.20 to show that if $0 \leq p \leq 1$, and $k = \lfloor np \rfloor$ (i.e., k is the largest integer less than or equal to np), then

$$\lim_{n\to\infty} \frac{1}{n} \log \binom{n}{k} = -p \log p - (1-p) \log(1-p) = H(p). \tag{11.330}$$

Now let p_i, $i = 1, \ldots, m$ be a probability distribution on m symbols (i.e., $p_i \geq 0$ and $\sum_i p_i = 1$). What is the limiting value of

$$\frac{1}{n} \log \binom{n}{\lfloor np_1 \rfloor \ \lfloor np_2 \rfloor \ \ldots \lfloor np_{m-1} \rfloor \ n - \sum_{j=0}^{m-1} \lfloor np_j \rfloor} = \frac{1}{n} \log \frac{n!}{\lfloor np_1 \rfloor! \ \lfloor np_2 \rfloor! \ \ldots \lfloor np_{m-1} \rfloor! \ (n - \sum_{j=0}^{m-1} \lfloor np_j \rfloor)!}? \tag{11.331}$$

11.22 *Running difference*. Let $X_1, X_2, \ldots, X_n$ be i.i.d. $\sim Q_1(x)$, and $Y_1, Y_2, \ldots, Y_n$ be i.i.d. $\sim Q_2(y)$. Let X^n and Y^n be independent. Find an expression for $\Pr\{\sum_{i=1}^n X_i - \sum_{i=1}^n Y_i \geq nt\}$ good to first order in the exponent. Again, this answer can be left in parametric form.

11.23 *Large likelihoods*. Let $X_1, X_2, \ldots$ be i.i.d. $\sim Q(x)$, $x \in \{1, 2, \ldots, m\}$. Let $P(x)$ be some other probability mass function. We form the log likelihood ratio

$$\frac{1}{n} \log \frac{P^n(X_1, X_2, \ldots, X_n)}{Q^n(X_1, X_2, \ldots, X_n)} = \frac{1}{n} \sum_{i=1}^{n} \log \frac{P(X_i)}{Q(X_i)}$$

of the sequence X^n and ask for the probability that it exceeds a certain threshold. Specifically, find (to first order in the exponent)

$$Q^n \left(\frac{1}{n} \log \frac{P(X_1, X_2, \ldots, X_n)}{Q(X_1, X_2, \ldots, X_n)} > 0 \right).$$

There may be an undetermined parameter in the answer.

11.24 *Fisher information for mixtures*. Let $f_1(x)$ and $f_0(x)$ be two given probability densities. Let Z be Bernoulli(θ), where θ is unknown. Let $X \sim f_1(x)$ if $Z = 1$ and $X \sim f_0(x)$ if $Z = 0$.

(a) Find the density $f_\theta(x)$ of the observed X.

(b) Find the Fisher information $J(\theta)$.

(c) What is the Cramér–Rao lower bound on the mean-squared error of an unbiased estimate of θ?

(d) Can you exhibit an unbiased estimator of θ?

11.25 *Bent coins*. Let $\{X_i\}$ be iid $\sim Q$, where

$$Q(k) = \Pr(X_i = k) = \binom{m}{k} q^k (1-q)^{m-k} \quad \text{for } k = 0, 1, 2, \ldots, m.$$

Thus, the X_i's are iid $\sim$ Binomial(m, q). Show that as $n \to \infty$,

$$\Pr\left(X_1 = k \,\middle|\, \frac{1}{n} \sum_{i=1}^{n} X_i \geq \alpha \right) \to P^*(k),$$

where P^* is Binomial(m, λ) (i.e., $P^*(k) = \binom{m}{k} \lambda^k (1-\lambda)^{m-k}$ for some $\lambda \in [0, 1]$). Find λ.

11.26 *Conditional limiting distribution*

(a) Find the exact value of

$$\Pr\left\{ X_1 = 1 \,\middle|\, \frac{1}{n} \sum_{i=1}^{n} X_i = \frac{1}{4} \right\} \tag{11.332}$$

if $X_1, X_2, \ldots,$ are Bernoulli($\frac{2}{3}$) and n is a multiple of 4.

(b) Now let $X_i \epsilon \{-1, 0, 1\}$ and let $X_1, X_2 \ldots$ be i.i.d. uniform over $\{-1, 0, +1\}$. Find the limit of

$$\Pr\left\{ X_1 = +1 \,\middle|\, \frac{1}{n} \sum_{i=1}^{n} X_i^2 = \frac{1}{2} \right\} \tag{11.333}$$

for $n = 2k, k \to \infty$.

11.27 *Variational inequality*. Verify for positive random variables X that

$$\log E_P(X) = \sup_Q \left[E_Q(\log X) - D(Q||P) \right], \tag{11.334}$$

where $E_P(X) = \sum_x xP(x)$ and $D(Q||P) = \sum_x Q(x) \log \frac{Q(x)}{P(x)}$ and the supremum is over all $Q(x) \geq 0$, $\sum Q(x) = 1$. It is enough to extremize $J(Q) = E_Q \ln X - D(Q||P) + \lambda(\sum Q(x) - 1)$.

11.28 *Type constraints*

(a) Find constraints on the type P_{X^n} such that the sample variance $\overline{X_n^2} - (\overline{X}_n)^2 \leq \alpha$, where $\overline{X_n^2} = \frac{1}{n}\sum_{i=1}^n X_i^2$ and $\overline{X}_n = \frac{1}{n}\sum_{i=1}^n X_i$.

(b) Find the exponent in the probability $Q^n(\overline{X_n^2} - (\overline{X}_n)^2 \leq \alpha)$. You can leave the answer in parametric form.

11.29 *Uniform distribution on the simplex*. Which of these methods will generate a sample from the uniform distribution on the simplex $\{x \in R^n : x_i \geq 0, \quad \sum_{i=1}^n x_i = 1\}$?

(a) Let Y_i be i.i.d. uniform [0, 1] with $X_i = Y_i / \sum_{j=1}^n Y_j$.

(b) Let Y_i be i.i.d. exponentially distributed $\sim \lambda e^{-\lambda y}$, $y \geq 0$, with $X_i = Y_i / \sum_{j=1}^n Y_j$.

(c) (*Break stick into n parts*) Let $Y_1, Y_2, \ldots, Y_{n-1}$ be i.i.d. uniform [0, 1], and let X_i be the length of the ith interval.

HISTORICAL NOTES

The method of types evolved from notions of strong typicality; some of the ideas were used by Wolfowitz [566] to prove channel capacity theorems. The method was fully developed by Csiszár and Körner [149], who derived the main theorems of information theory from this viewpoint. The method of types described in Section 11.1 follows the development in Csiszár and Körner. The $\mathcal{L}_1$ lower bound on relative entropy is due to Csiszár [138], Kullback [336], and Kemperman [309]. Sanov's theorem [455] was generalized by Csiszár [141] using the method of types.

CHAPTER 12

MAXIMUM ENTROPY

The temperature of a gas corresponds to the average kinetic energy of the molecules in the gas. What can we say about the distribution of velocities in the gas at a given temperature? We know from physics that this distribution is the maximum entropy distribution under the temperature constraint, otherwise known as the Maxwell–Boltzmann distribution. The maximum entropy distribution corresponds to the macrostate (as indexed by the empirical distribution) that has the most microstates (the individual gas velocities). Implicit in the use of maximum entropy methods in physics is a sort of AEP which says that all microstates are equally probable.

12.1 MAXIMUM ENTROPY DISTRIBUTIONS

Consider the following problem: Maximize the entropy $h(f)$ over all probability densities f satisfying

$$\begin{array}{ll} 1. & f(x) \geq 0, \text{ with equality outside the support set } S \\ 2. & \int_S f(x)\,dx = 1 \\ 3. & \int_S f(x) r_i(x)\,dx = \alpha_i \quad \text{for } 1 \leq i \leq m. \end{array} \tag{12.1}$$

Thus, f is a density on support set S meeting certain moment constraints $\alpha_1, \alpha_2, \ldots, \alpha_m$.

Approach 1 (Calculus) The differential entropy $h(f)$ is a concave function over a convex set. We form the functional

$$J(f) = -\int f \ln f + \lambda_0 \int f + \sum_{i=1}^{m} \lambda_i \int f r_i \tag{12.2}$$

and "differentiate" with respect to $f(x)$, the xth component of f, to obtain

$$\frac{\partial J}{\partial f(x)} = -\ln f(x) - 1 + \lambda_0 + \sum_{i=1}^{m} \lambda_i r_i(x). \tag{12.3}$$

Elements of Information Theory, Second Edition, By Thomas M. Cover and Joy A. Thomas

Setting this equal to zero, we obtain the form of the maximizing density

$$f(x) = e^{\lambda_0 - 1 + \sum_{i=1}^{m} \lambda_i r_i(x)}, \qquad x \in S, \tag{12.4}$$

where $\lambda_0, \lambda_1, \ldots, \lambda_m$ are chosen so that f satisfies the constraints.

The approach using calculus only suggests the form of the density that maximizes the entropy. To prove that this is indeed the maximum, we can take the second variation. It is simpler to use the information inequality $D(g||f) \geq 0$.

Approach 2 *(Information inequality)* If g satisfies (12.1) and if f^* is of the form (12.4), then $0 \leq D(g||f^*) = -h(g) + h(f^*)$. Thus $h(g) \leq h(f^*)$ for all g satisfying the constraints. We prove this in the following theorem.

Theorem 12.1.1 (*Maximum entropy distribution*) *Let* $f^*(x) = f_\lambda(x) = e^{\lambda_0 + \sum_{i=1}^{m} \lambda_i r_i(x)}$, $x \in S$, *where* $\lambda_0, \ldots, \lambda_m$ *are chosen so that* f^* *satisfies (12.1). Then* f^* *uniquely maximizes* $h(f)$ *over all probability densities* f *satisfying constraints (12.1).*

Proof: Let g satisfy the constraints (12.1). Then

$$h(g) = -\int_S g \ln g \tag{12.5}$$

$$= -\int_S g \ln \frac{g}{f^*} f^* \tag{12.6}$$

$$= -D(g||f^*) - \int_S g \ln f^* \tag{12.7}$$

$$\overset{(a)}{\leq} -\int_S g \ln f^* \tag{12.8}$$

$$\overset{(b)}{=} -\int_S g \left(\lambda_0 + \sum \lambda_i r_i\right) \tag{12.9}$$

$$\overset{(c)}{=} -\int_S f^* \left(\lambda_0 + \sum \lambda_i r_i\right) \tag{12.10}$$

$$= -\int_S f^* \ln f^* \tag{12.11}$$

$$= h(f^*), \tag{12.12}$$

where (a) follows from the nonnegativity of relative entropy, (b) follows from the definition of f^*, and (c) follows from the fact that both f^* and g satisfy the constraints. Note that equality holds in (a) if and only

if $g(x) = f^*(x)$ for all x, except for a set of measure 0, thus proving uniqueness. □

The same approach holds for discrete entropies and for multivariate distributions.

12.2 EXAMPLES

Example 12.2.1 (*One-dimensional gas with a temperature constraint*) Let the constraints be $EX = 0$ and $EX^2 = \sigma^2$. Then the form of the maximizing distribution is

$$f(x) = e^{\lambda_0 + \lambda_1 x + \lambda_2 x^2}. \tag{12.13}$$

To find the appropriate constants, we first recognize that this distribution has the same form as a normal distribution. Hence, the density that satisfies the constraints and also maximizes the entropy is the $\mathcal{N}(0, \sigma^2)$ distribution:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}. \tag{12.14}$$

Example 12.2.2 (*Dice, no constraints*) Let $S = \{1, 2, 3, 4, 5, 6\}$. The distribution that maximizes the entropy is the uniform distribution, $p(x) = \frac{1}{6}$ for $x \in S$.

Example 12.2.3 (*Dice, with $EX = \sum ip_i = \alpha$*) This important example was used by Boltzmann. Suppose that n dice are thrown on the table and we are told that the total number of spots showing is $n\alpha$. What proportion of the dice are showing face i, $i = 1, 2, \ldots, 6$?

One way of going about this is to count the number of ways that n dice can fall so that n_i dice show face i. There are $\binom{n}{n_1, n_2, \ldots, n_6}$ such ways. This is a macrostate indexed by $(n_1, n_2, \ldots, n_6)$ corresponding to $\binom{n}{n_1, n_2, \ldots, n_6}$ microstates, each having probability $\frac{1}{6^n}$. To find the most probable macrostate, we wish to maximize $\binom{n}{n_1, n_2, \ldots, n_6}$ under the constraint observed on the total number of spots,

$$\sum_{i=1}^{6} in_i = n\alpha. \tag{12.15}$$

Using a crude Stirling's approximation, $n! \approx (\frac{n}{e})^n$, we find that

$$\binom{n}{n_1, n_2, \ldots, n_6} \approx \frac{(\frac{n}{e})^n}{\prod_{i=1}^{6} (\frac{n_i}{e})^{n_i}} \tag{12.16}$$

$$= \prod_{i=1}^{6} \left(\frac{n}{n_i} \right)^{n_i} \tag{12.17}$$

$$= e^{nH\left(\frac{n_1}{n}, \frac{n_2}{n}, \ldots, \frac{n_6}{n}\right)}. \tag{12.18}$$

Thus, maximizing $\binom{n}{n_1, n_2, \ldots, n_6}$ under the constraint (12.15) is almost equivalent to maximizing $H(p_1, p_2, \ldots, p_6)$ under the constraint $\sum ip_i = \alpha$. Using Theorem 12.1.1 under this constraint, we find the maximum entropy probability mass function to be

$$p_i^* = \frac{e^{\lambda i}}{\sum_{i=1}^{6} e^{\lambda i}}, \tag{12.19}$$

where λ is chosen so that $\sum ip_i^* = \alpha$. Thus, the most probable macrostate is $(np_1^*, np_2^* \ldots, np_6^*)$, and we expect to find $n_i^* = np_i^*$ dice showing face i.

In Chapter 11 we show that the reasoning and the approximations are essentially correct. In fact, we show that not only is the maximum entropy macrostate the most likely, but it also contains almost all of the probability. Specifically, for rational α,

$$\Pr\left\{ \left| \frac{N_i}{n} - p_i^* \right| < \epsilon, i = 1, 2, \ldots, 6 \,\middle|\, \sum_{i=1}^{n} X_i = n\alpha \right\} \to 1, \tag{12.20}$$

as $n \to \infty$ along the subsequence such that $n\alpha$ is an integer.

Example 12.2.4 Let $S = [a, b]$, with no other constraints. Then the maximum entropy distribution is the uniform distribution over this range.

Example 12.2.5 $S = [0, \infty)$ and $EX = \mu$. Then the entropy-maximizing distribution is

$$f(x) = \frac{1}{\mu} e^{-\frac{x}{\mu}}, \quad x \geq 0. \tag{12.21}$$

This problem has a physical interpretation. Consider the distribution of the height X of molecules in the atmosphere. The average potential energy of the molecules is fixed, and the gas tends to the distribution that has the maximum entropy subject to the constraint that $E(mgX)$ is fixed. This is the exponential distribution with density $f(x) = \lambda e^{-\lambda x}$, $x \geq 0$. The density of the atmosphere does indeed have this distribution.

Example 12.2.6 $S = (-\infty, \infty)$, and $EX = \mu$. Here the maximum entropy is infinite, and there is no maximum entropy distribution. (Consider normal distributions with larger and larger variances.)

Example 12.2.7 $S = (-\infty, \infty)$, $EX = \alpha_1$, and $EX^2 = \alpha_2$. The maximum entropy distribution is $\mathcal{N}(\alpha_1, \alpha_2 - \alpha_1^2)$.

Example 12.2.8 $S = \mathcal{R}^n$, $EX_iX_j = K_{ij}$, $1 \le i, j \le n$. This is a multivariate example, but the same analysis holds and the maximum entropy density is of the form

$$f(\mathbf{x}) = e^{\lambda_0 + \sum_{i,j} \lambda_{ij} x_i x_j}. \tag{12.22}$$

Since the exponent is a quadratic form, it is clear by inspection that the density is a multivariate normal with zero mean. Since we have to satisfy the second moment constraints, we must have a multivariate normal with covariance K_{ij}, and hence the density is

$$f(\mathbf{x}) = \frac{1}{\left(\sqrt{2\pi}\right)^n |K|^{1/2}} e^{-\frac{1}{2}\mathbf{x}^T K^{-1}\mathbf{x}}, \tag{12.23}$$

which has an entropy

$$h(\mathcal{N}_n(0, K)) = \frac{1}{2}\log(2\pi e)^n |K|, \tag{12.24}$$

as derived in Chapter 8.

Example 12.2.9 Suppose that we have the same constraints as in Example 12.2.8, but $EX_iX_j = K_{ij}$ only for some restricted set of $(i, j) \in A$. For example, we might know only K_{ij} for $i = j \pm 2$. Then by comparing (12.22) and (12.23), we can conclude that $(K^{-1})_{ij} = 0$ for $(i, j) \in A^c$ (i.e., the entries in the inverse of the covariance matrix are 0 when (i, j) is outside the constraint set).

12.3 ANOMALOUS MAXIMUM ENTROPY PROBLEM

We have proved that the maximum entropy distribution subject to the constraints

$$\int_S h_i(x) f(x)\, dx = \alpha_i \tag{12.25}$$

is of the form

$$f(x) = e^{\lambda_0 + \sum \lambda_i h_i(x)} \tag{12.26}$$

if $\lambda_0, \lambda_1, \ldots, \lambda_p$ satisfying the constraints (12.25) exist.

We now consider a tricky problem in which the λ_i cannot be chosen to satisfy the constraints. Nonetheless, the "maximum" entropy can be found. We consider the following problem: Maximize the entropy subject to the constraints

$$\int_{-\infty}^{\infty} f(x)\,dx = 1, \tag{12.27}$$

$$\int_{-\infty}^{\infty} xf(x)\,dx = \alpha_1, \tag{12.28}$$

$$\int_{-\infty}^{\infty} x^2 f(x)\,dx = \alpha_2, \tag{12.29}$$

$$\int_{-\infty}^{\infty} x^3 f(x)\,dx = \alpha_3. \tag{12.30}$$

Here, the maximum entropy distribution, if it exists, must be of the form

$$f(x) = e^{\lambda_0+\lambda_1 x+\lambda_2 x^2+\lambda_3 x^3}. \tag{12.31}$$

But if λ_3 is nonzero, $\int_{-\infty}^{\infty} f = \infty$ and the density cannot be normalized. So λ_3 must be 0. But then we have four equations and only three variables, so that in general it is not possible to choose the appropriate constants. The method seems to have failed in this case.

The reason for the apparent failure is simple: The entropy has a least upper bound under these constraints, but it is not possible to attain it. Consider the corresponding problem with only first and second moment constraints. In this case, the results of Example 12.2.1 show that the entropy-maximizing distribution is the normal with the appropriate moments. With the additional third moment constraint, the maximum entropy cannot be higher. Is it possible to achieve this value?

We cannot achieve it, but we can come arbitrarily close. Consider a normal distribution with a small "wiggle" at a very high value of x. The moments of the new distribution are almost the same as those of the old one, the biggest change being in the third moment. We can bring the first and second moments back to their original values by adding new wiggles to balance out the changes caused by the first. By choosing the position of the wiggles, we can get any value of the third moment without reducing the entropy significantly below that of the associated normal. Using this method, we can come arbitrarily close to the upper bound for the maximum entropy distribution. We conclude that

$$\sup h(f) = h(\mathcal{N}(0, \alpha_2 - \alpha_1^2)) = \frac{1}{2}\ln 2\pi e(\alpha_2 - \alpha_1^2). \tag{12.32}$$

This example shows that the maximum entropy may only be ϵ-achievable.

12.4 SPECTRUM ESTIMATION

Given a stationary zero-mean stochastic process $\{X_i\}$, we define the autocorrelation function as

$$R(k) = EX_iX_{i+k}. \tag{12.33}$$

The Fourier transform of the autocorrelation function for a zero-mean process is the power spectral density $S(\lambda)$:

$$S(\lambda) = \sum_{m=-\infty}^{\infty} R(m)e^{-im\lambda}, \quad -\pi < \lambda \leq \pi, \tag{12.34}$$

where $i = \sqrt{-1}$. Since the power spectral density is indicative of the structure of the process, it is useful to form an estimate from a sample of the process.

There are many methods to estimate the power spectrum. The simplest way is to estimate the autocorrelation function by taking sample averages for a sample of length n,

$$\hat{R}(k) = \frac{1}{n-k} \sum_{i=1}^{n-k} X_iX_{i+k}. \tag{12.35}$$

If we use all the values of the sample correlation function $\hat{R}(\cdot)$ to calculate the spectrum, the estimate that we obtain from (12.34) does not converge to the true power spectrum for large n. Hence, this method, the *periodogram method*, is rarely used. One of the reasons for the problem with the periodogram method is that the estimates of the autocorrelation function from the data have different accuracies. The estimates for low values of k (called the *lags*) are based on a large number of samples and those for high k on very few samples. So the estimates are more accurate at low k. The method can be modified so that it depends only on the autocorrelations at low k by setting the higher lag autocorrelations to 0. However, this introduces some artifacts because of the sudden transition to zero autocorrelation. Various windowing schemes have been suggested to smooth out the transition. However, windowing reduces spectral resolution and can give rise to negative power spectral estimates.

In the late 1960s, while working on the problem of spectral estimation for geophysical applications, Burg suggested an alternative method. Instead of

setting the autocorrelations at high lags to zero, he set them to values that make the fewest assumptions about the data (i.e., values that maximize the entropy rate of the process). This is consistent with the maximum entropy principle as articulated by Jaynes [294]. Burg assumed the process to be stationary and Gaussian and found that the process which maximizes the entropy subject to the correlation constraints is an autoregressive Gaussian process of the appropriate order. In some applications where we can assume an underlying autoregressive model for the data, this method has proved useful in determining the parameters of the model (e.g., linear predictive coding for speech). This method (known as the *maximum entropy method* or *Burg's method*) is a popular method for estimation of spectral densities. We prove Burg's theorem in Section 12.6.

12.5 ENTROPY RATES OF A GAUSSIAN PROCESS

In Chapter 8 we defined the differential entropy of a continuous random variable. We can now extend the definition of entropy rates to real-valued stochastic processes.

Definition The *differential entropy rate* of a stochastic process $\{X_i\}$, $X_i \in \mathcal{R}$, is defined to be

$$h(\mathcal{X}) = \lim_{n\to\infty} \frac{h(X_1, X_2, \ldots, X_n)}{n} \tag{12.36}$$

if the limit exists.

Just as in the discrete case, we can show that the limit exists for stationary processes and that the limit is given by the two expressions

$$h(\mathcal{X}) = \lim_{n\to\infty} \frac{h(X_1, X_2, \ldots, X_n)}{n} \tag{12.37}$$

$$= \lim_{n\to\infty} h(X_n|X_{n-1}, \ldots, X_1). \tag{12.38}$$

For a stationary Gaussian stochastic process, we have

$$h(X_1, X_2, \ldots, X_n) = \frac{1}{2}\log(2\pi e)^n |K^{(n)}|, \tag{12.39}$$

where the covariance matrix $K^{(n)}$ is Toeplitz with entries $R(0), R(1), \ldots, R(n-1)$ along the top row. Thus, $K_{ij}^{(n)} = R(i-j) = E(X_i - EX_i)(X_j$

$- EX_j$). As $n \to \infty$, the density of the eigenvalues of the covariance matrix tends to a limit, which is the spectrum of the stochastic process. Indeed, Kolmogorov showed that the entropy rate of a stationary Gaussian stochastic process can be expressed as

$$h(\mathcal{X}) = \frac{1}{2} \log 2\pi e + \frac{1}{4\pi} \int_{-\pi}^{\pi} \log S(\lambda)\, d\lambda. \tag{12.40}$$

The entropy rate is also $\lim_{n\to\infty} h(X_n|X^{n-1})$. Since the stochastic process is Gaussian, the conditional distribution is also Gaussian, and hence the conditional entropy is $\frac{1}{2} \log 2\pi e \sigma_\infty^2$, where σ_∞^2 is the variance of the error in the best estimate of X_n given the infinite past. Thus,

$$\sigma_\infty^2 = \frac{1}{2\pi e} 2^{2h(\mathcal{X})}, \tag{12.41}$$

where $h(\mathcal{X})$ is given by (12.40). Hence, the entropy rate corresponds to the minimum mean-squared error of the best estimator of a sample of the process given the infinite past.

12.6 BURG'S MAXIMUM ENTROPY THEOREM

Theorem 12.6.1 *The maximum entropy rate stochastic process* $\{X_i\}$ *satisfying the constraints*

$$EX_i X_{i+k} = \alpha_k, \quad k = 0, 1, \ldots, p \quad \text{for all } i, \tag{12.42}$$

is the pth-order Gauss–Markov process of the form

$$X_i = -\sum_{k=1}^{p} a_k X_{i-k} + Z_i, \tag{12.43}$$

where the Z_i *are i.i.d.* $\sim \mathcal{N}(0, \sigma^2)$ *and* $a_1, a_2, \ldots, a_p, \sigma^2$ *are chosen to satisfy (12.42).*

Remark We do not assume that $\{X_i\}$ is (a) zero mean, (b) Gaussian, or (c) wide-sense stationary.

Proof: Let $X_1, X_2, \ldots, X_n$ be any stochastic process that satisfies the constraints (12.42). Let $Z_1, Z_2, \ldots, Z_n$ be a Gaussian process with the same covariance matrix as $X_1, X_2, \ldots, X_n$. Then since the multivariate normal distribution maximizes the entropy over all vector-valued random

variables under a covariance constraint, we have

$$h(X_1, X_2, \ldots, X_n) \leq h(Z_1, Z_2, \ldots, Z_n) \tag{12.44}$$

$$= h(Z_1, \ldots, Z_p) + \sum_{i=p+1}^{n} h(Z_i|Z_{i-1}, Z_{i-2}, \ldots, Z_1) \tag{12.45}$$

$$\leq h(Z_1, \ldots, Z_p) + \sum_{i=p+1}^{n} h(Z_i|Z_{i-1}, Z_{i-2}, \ldots, Z_{i-p}) \tag{12.46}$$

by the chain rule and the fact that conditioning reduces entropy. Now define $Z_1', Z_2', \ldots, Z_n'$ as a pth-order Gauss–Markov process with the same distribution as $Z_1, Z_2, \ldots, Z_n$ for all orders up to p. (Existence of such a process will be verified using the Yule–Walker equations immediately after the proof.) Then since $h(Z_i|Z_{i-1}, \ldots, Z_{i-p})$ depends only on the pth-order distribution, $h(Z_i|Z_{i-1}, \ldots, Z_{i-p}) = h(Z_i'|Z_{i-1}', \ldots, Z_{i-p}')$, and continuing the chain of inequalities, we obtain

$$h(X_1, X_2, \ldots, X_n) \leq h(Z_1, \ldots, Z_p) + \sum_{i=p+1}^{n} h(Z_i|Z_{i-1}, Z_{i-2}, \ldots, Z_{i-p}) \tag{12.47}$$

$$= h(Z_1', \ldots, Z_p') + \sum_{i=p+1}^{n} h(Z_i'|Z_{i-1}', Z_{i-2}', \ldots, Z_{i-p}') \tag{12.48}$$

$$= h(Z_1', Z_2', \ldots, Z_n'), \tag{12.49}$$

where the last equality follows from the pth-order Markovity of the $\{Z_i'\}$. Dividing by n and taking the limit, we obtain

$$\overline{\lim} \frac{1}{n} h(X_1, X_2, \ldots, X_n) \leq \lim \frac{1}{n} h(Z_1', Z_2', \ldots, Z_n') = h^*, \tag{12.50}$$

where

$$h^* = \frac{1}{2} \log 2\pi e \sigma^2, \tag{12.51}$$

which is the entropy rate of the Gauss–Markov process. Hence, the maximum entropy rate stochastic process satisfying the constraints is the pth-order Gauss–Markov process satisfying the constraints. □

A bare-bones summary of the proof is that the entropy of a finite segment of a stochastic process is bounded above by the entropy of a

segment of a Gaussian random process with the same covariance structure. This entropy is in turn bounded above by the entropy of the minimal order Gauss–Markov process satisfying the given covariance constraints. Such a process exists and has a convenient characterization by means of the Yule–Walker equations given below.

Note on the choice of $a_1, \ldots, a_p$ *and* σ^2: Given a sequence of covariances $R(0), R(1), \ldots, R(p)$, does there exist a pth-order Gauss–Markov process with these covariances? Given a process of the form (12.43), can we choose the a_k's to satisfy the constraints? Multiplying (12.43) by X_{i-l} and taking expectations, noting that $R(k) = R(-k)$, we get

$$R(0) = -\sum_{k=1}^{p} a_k R(-k) + \sigma^2 \tag{12.52}$$

and

$$R(l) = -\sum_{k=1}^{p} a_k R(l-k), \quad l = 1, 2, \ldots. \tag{12.53}$$

These equations are called the *Yule–Walker equations*. There are $p+1$ equations in the $p+1$ unknowns $a_1, a_2, \ldots, a_p, \sigma^2$. Therefore, we can solve for the parameters of the process from the covariances.

Fast algorithms such as the Levinson algorithm and the Durbin algorithm [433] have been devised to use the special structure of these equations to calculate the coefficients $a_1, a_2, \ldots, a_p$ efficiently from the covariances. (We set $a_0 = 1$ for a consistent notation.) Not only do the Yule–Walker equations provide a convenient set of linear equations for calculating the a_k's and σ^2 from the $R(k)$'s, they also indicate how the autocorrelations behave for lags greater than p. The autocorrelations for high lags are an extension of the values for lags less than p. These values are called the Yule–Walker extension of the autocorrelations. The spectrum of the maximum entropy process is seen to be

$$S(\lambda) = \sum_{m=-\infty}^{\infty} R(m) e^{-im\lambda} \tag{12.54}$$

$$= \frac{\sigma^2}{\left|1 + \sum_{k=1}^{p} a_k e^{-ik\lambda}\right|^2}, \quad -\pi \le \lambda \le \pi. \tag{12.55}$$

This is the maximum entropy spectral density subject to the constraints $R(0), R(1), \ldots, R(p)$.

However, for the pth-order Gauss–Markov process, it is possible to calculate the entropy rate directly without calculating the a_i's. Let K_p be

the autocorrelation matrix corresponding to this process—the matrix with $R_0, R_1, \ldots, R_p$ along the top row. For this process, the entropy rate is equal to

$$h^* = h(X_p|X_{p-1}, \ldots, X_0) = h(X_0, \ldots, X_p) - h(X_0, \ldots, X_{p-1}) \tag{12.56}$$

$$= \frac{1}{2}\log(2\pi e)^{p+1}|K_p| - \frac{1}{2}\log(2\pi e)^p|K_{p-1}| \tag{12.57}$$

$$= \frac{1}{2}\log(2\pi e)\frac{|K_p|}{|K_{p-1}|}. \tag{12.58}$$

In a practical problem, we are generally given a sample sequence $X_1, X_2, \ldots, X_n$, from which we calculate the autocorrelations. An important question is: How many autocorrelation lags should we consider (i.e., what is the optimum value of p)? A logically sound method is to choose the value of p that minimizes the total description length in a two-stage description of the data. This method has been proposed by Rissanen [442, 447] and Barron [33] and is closely related to the idea of Kolmogorov complexity.

SUMMARY

Maximum entropy distribution. Let f be a probability density satisfying the constraints

$$\int_S f(x)r_i(x) = \alpha_i \quad \text{for } 1 \le i \le m. \tag{12.59}$$

Let $f^*(x) = f_\lambda(x) = e^{\lambda_0 + \sum_{i=1}^m \lambda_i r_i(x)}$, $x \in S$, and let $\lambda_0, \ldots, \lambda_m$ be chosen so that f^* satisfies (12.59). Then f^* uniquely maximizes $h(f)$ over all f satisfying these constraints.

Maximum entropy spectral density estimation. The entropy rate of a stochastic process subject to autocorrelation constraints $R_0, R_1, \ldots, R_p$ is maximized by the pth order zero-mean Gauss-Markov process satisfying these constraints. The maximum entropy rate is

$$h^* = \frac{1}{2}\log(2\pi e)\frac{|K_p|}{|K_{p-1}|}, \tag{12.60}$$

and the maximum entropy spectral density is

$$S(\lambda) = \frac{\sigma^2}{\left|1 + \sum_{k=1}^{p} a_k e^{-ik\lambda}\right|^2}. \tag{12.61}$$

PROBLEMS

12.1 *Maximum entropy*. Find the maximum entropy density f, defined for $x \geq 0$, satisfying $EX = \alpha_1$, $E \ln X = \alpha_2$. That is, maximize $-\int f \ln f$ subject to $\int x f(x)\, dx = \alpha_1$, $\int (\ln x) f(x)\, dx = \alpha_2$, where the integral is over $0 \leq x < \infty$. What family of densities is this?

12.2 *Min $D(P \| Q)$ under constraints on P*. We wish to find the (parametric form) of the probability mass function $P(x)$, $x \in \{1, 2, \ldots\}$ that minimizes the relative entropy $D(P \| Q)$ over all P such that $\sum P(x) g_i(x) = \alpha_i$, $i = 1, 2, \ldots$.

(a) Use Lagrange multipliers to guess that

$$P^*(x) = Q(x) e^{\sum_{i=1}^{\infty} \lambda_i g_i(x) + \lambda_0} \tag{12.62}$$

achieves this minimum if there exist λ_i's satisfying the α_i constraints. This generalizes the theorem on maximum entropy distributions subject to constraints.

(b) Verify that P^* minimizes $D(P \| Q)$.

12.3 *Maximum entropy processes*. Find the maximum entropy rate stochastic process $\{X_i\}_{-\infty}^{\infty}$ subject to the constraints:

(a) $EX_i^2 = 1, \quad i = 1, 2, \ldots$.

(b) $EX_i^2 = 1$, $EX_i X_{i+1} = \frac{1}{2}, \quad i = 1, 2, \ldots$.

(c) Find the maximum entropy spectrum for the processes in parts (a) and (b).

12.4 *Maximum entropy with marginals*. What is the maximum entropy distribution $p(x, y)$ that has the following marginals?

x \ y	1	2	3	
1	p_{11}	p_{12}	p_{13}	$\frac{1}{2}$
2	p_{21}	p_{22}	p_{23}	$\frac{1}{4}$
3	p_{31}	p_{32}	p_{33}	$\frac{1}{4}$
	$\frac{2}{3}$	$\frac{1}{6}$	$\frac{1}{6}$	

(*Hint:* You may wish to guess and verify a more general result.)

12.5 *Processes with fixed marginals*. Consider the set of all densities with fixed pairwise marginals $f_{X_1,X_2}(x_1, x_2)$, $f_{X_2,X_3}(x_2, x_3), \ldots,$ $f_{X_{n-1},X_n}(x_{n-1}, x_n)$. Show that the maximum entropy process with these marginals is the first-order (possibly time-varying) Markov process with these marginals. Identify the maximizing $f^*(x_1, x_2, \ldots, x_n)$.

12.6 *Every density is a maximum entropy density*. Let $f_0(x)$ be a given density. Given $r(x)$, let $g_\alpha(x)$ be the density maximizing $h(X)$ over all f satisfying $\int f(x)r(x)\,dx = \alpha$. Now let $r(x) = \ln f_0(x)$. Show that $g_\alpha(x) = f_0(x)$ for an appropriate choice $\alpha = \alpha_0$. Thus, $f_0(x)$ is a maximum entropy density under the constraint $\int f \ln f_0 = \alpha_0$.

12.7 *Mean-squared error*. Let $\{X_i\}_{i=1}^n$ satisfy $EX_iX_{i+k} = R_k, k = 0, 1, \ldots, p$. Consider linear predictors for X_n; that is,

$$\hat{X}_n = \sum_{i=1}^{n-1} b_i X_{n-i}.$$

Assume that $n > p$. Find

$$\max_{f(x^n)} \min_{b} E(X_n - \hat{X}_n)^2,$$

where the minimum is over all linear predictors b and the maximum is over all densities f satisfying $R_0, \ldots, R_p$.

12.8 *Maximum entropy characteristic functions*. We ask for the maximum entropy density $f(x), 0 \le x \le a$, satisfying a constraint on the characteristic function $\Psi(u) = \int_0^a e^{iux} f(x)\,dx$. The answers need be given only in parametric form.

(a) Find the maximum entropy f satisfying $\int_0^a f(x)\cos(u_0x)\,dx = \alpha$, at a specified point u_0.

(b) Find the maximum entropy f satisfying $\int_0^a f(x)\sin(u_0x)\,dx = \beta$.

(c) Find the maximum entropy density $f(x), 0 \le x \le a$, having a given value of the characteristic function $\Psi(u_0)$ at a specified point u_0.

(d) What problem is encountered if $a = \infty$?

12.9 *Maximum entropy processes*

(a) Find the maximum entropy rate binary stochastic process $\{X_i\}_{i=-\infty}^{\infty}$, $X_i \in \{0, 1\}$, satisfying $\Pr\{X_i = X_{i+1}\} = \frac{1}{3}$ for all i.

(b) What is the resulting entropy rate?

12.10 *Maximum entropy of sums*. Let $Y = X_1 + X_2$. Find the maximum entropy density for Y under the constraint $EX_1^2 = P_1$, $EX_2^2 = P_2$:

(a) If X_1 and X_2 are independent.

(b) If X_1 and X_2 are allowed to be dependent.

(c) Prove part (a).

12.11 *Maximum entropy Markov chain*. Let $\{X_i\}$ be a stationary Markov chain with $X_i \in \{1, 2, 3\}$. Let $I(X_n; X_{n+2}) = 0$ for all n.

(a) What is the maximum entropy rate process satisfying this constraint?

(b) What if $I(X_n; X_{n+2}) = \alpha$ for all n for some given value of α, $0 \leq \alpha \leq \log 3$?

12.12 *Entropy bound on prediction error*. Let $\{X_n\}$ be an arbitrary real valued stochastic process. Let $\hat{X}_{n+1} = E\{X_{n+1}|X^n\}$. Thus the conditional mean $\hat{X}_{n+1}$ is a random variable depending on the n-past X^n. Here $\hat{X}_{n+1}$ is the minimum mean squared error prediction of X_{n+1} given the past.

(a) Find a lower bound on the conditional variance $E\{E\{(X_{n+1} - \hat{X}_{n+1})^2|X^n\}\}$ in terms of the conditional differential entropy $h(X_{n+1}|X^n)$.

(b) Is equality achieved when $\{X_n\}$ is a Gaussian stochastic process?

12.13 *Maximum entropy rate*. What is the maximum entropy rate stochastic process $\{X_i\}$ over the symbol set $\{0, 1\}$ for which the probability that 00 occurs in a sequence is zero?

12.14 *Maximum entropy*

(a) What is the parametric-form maximum entropy density $f(x)$ satisfying the two conditions

$$EX^8 = a, \qquad EX^{16} = b?$$

(b) What is the maximum entropy density satisfying the condition

$$E(X^8 + X^{16}) = a + b?$$

(c) Which entropy is higher?

12.15 *Maximum entropy*. Find the parametric form of the maximum entropy density f satisfying the Laplace transform condition

$$\int f(x)\mathrm{e}^{-x}\,dx = \alpha,$$

and give the constraints on the parameter.

12.16 *Maximum entropy processes*. Consider the set of all stochastic processes with $\{X_i\}$, $X_i \in \mathcal{R}$, with

$$R_0 = EX_i^2 = 1, \qquad R_1 = EX_iX_{i+1} = \frac{1}{2}.$$

Find the maximum entropy rate.

12.17 *Binary maximum entropy*. Consider a *binary* process $\{X_i\}$, $X_i \in \{-1, +1\}$, with $R_0 = EX_i^2 = 1$ and $R_1 = EX_iX_{i+1} = \frac{1}{2}$.

(a) Find the maximum entropy process with these constraints.

(b) What is the entropy rate?

(c) Is there a Bernoulli process satisfying these constraints?

12.18 *Maximum entropy*. Maximize $h(Z, V_x, V_y, V_z)$ subject to the energy constraint $E(\frac{1}{2}m\|V\|^2 + mgZ) = E_0$. Show that the resulting distribution yields

$$E\frac{1}{2}m\|V\|^2 = \frac{3}{5}E_0, \qquad EmgZ = \frac{2}{5}E_0.$$

Thus, $\frac{2}{5}$ of the energy is stored in the potential field, regardless of its strength g.

12.19 *Maximum entropy discrete processes*

(a) Find the maximum entropy rate binary stochastic process $\{X_i\}_{i=-\infty}^{\infty}$, $X_i \in \{0, 1\}$, satisfying $\Pr\{X_i = X_{i+1}\} = \frac{1}{3}$ for all i.

(b) What is the resulting entropy rate?

12.20 *Maximum entropy of sums*. Let $Y = X_1 + X_2$. Find the maximum entropy of Y under the constraint $EX_1^2 = P_1$, $EX_2^2 = P_2$:

(a) If X_1 and X_2 are independent.

(b) If X_1 and X_2 are allowed to be dependent.

12.21 *Entropy rate*

(a) Find the maximum entropy rate stochastic process $\{X_i\}$ with $EX_i^2 = 1,\ EX_iX_{i+2} = \alpha,\ i = 1, 2, \ldots$. Be careful.

(b) What is the maximum entropy rate?

(c) What is EX_iX_{i+1} for this process?

12.22 *Minimum expected value*

(a) Find the minimum value of EX over all probability density functions $f(x)$ satisfying the following three constraints:

(i) $f(x) = 0$ for $x \leq 0$.

(ii) $\int_{-\infty}^{\infty} f(x)\,dx = 1$.

(iii) $h(f) = h$.

(b) Solve the same problem if (i) is replaced by

(i′) $f(x) = 0$ for $x \leq a$.

HISTORICAL NOTES

The maximum entropy principle arose in statistical mechanics in the nineteenth century and has been advocated for use in a broader context by Jaynes [294]. It was applied to spectral estimation by Burg [80]. The information-theoretic proof of Burg's theorem is from Choi and Cover [98].

CHAPTER 13

UNIVERSAL SOURCE CODING

Here we develop the basics of universal source coding. Minimax regret data compression is defined, and the descriptive cost of universality is shown to be the information radius of the relative entropy ball containing all the source distributions. The minimax theorem shows this radius to be the channel capacity for the associated channel given by the source distribution. Arithmetic coding enables the use of a source distribution that is learned on the fly. Finally, individual sequence compression is defined and achieved by a succession of Lempel–Ziv parsing algorithms.

In Chapter 5 we introduced the problem of finding the shortest representation of a source, and showed that the entropy is the fundamental lower limit on the expected length of any uniquely decodable representation. We also showed that if we know the probability distribution for the source, we can use the Huffman algorithm to construct the optimal (minimal expected length) code for that distribution.

For many practical situations, however, the probability distribution underlying the source may be unknown, and we cannot apply the methods of Chapter 5 directly. Instead, all we know is a class of distributions. One possible approach is to wait until we have seen all the data, estimate the distribution from the data, use this distribution to construct the best code, and then go back to the beginning and compress the data using this code. This two-pass procedure is used in some applications where there is a fairly small amount of data to be compressed. But there are many situations in which it is not feasible to make two passes over the data, and it is desirable to have a one-pass (or online) algorithm to compress the data that "learns" the probability distribution of the data and uses it to compress the incoming symbols. We show the existence of such algorithms that do well for any distribution within a class of distributions.

In yet other cases, there is no probability distribution underlying the data—all we are given is an individual sequence of outcomes. Examples

Elements of Information Theory, Second Edition, By Thomas M. Cover and Joy A. Thomas

of such data sources include text and music. We can then ask the question: How well can we compress the sequence? If we do not put any restrictions on the class of algorithms, we get a meaningless answer—there always exists a function that compresses a particular sequence to one bit while leaving every other sequence uncompressed. This function is clearly "overfitted" to the data. However, if we compare our performance to that achievable by optimal word assignments with respect to Bernoulli distributions or kth-order Markov processes, we obtain more interesting answers that are in many ways analogous to the results for the probabilistic or average case analysis. The ultimate answer for compressibility for an individual sequence is the Kolmogorov complexity of the sequence, which we discuss in Chapter 14.

We begin the chapter by considering the problem of source coding as a game in which the coder chooses a code that attempts to minimize the average length of the representation and nature chooses a distribution on the source sequence. We show that this game has a value that is related to the capacity of a channel with rows of its transition matrix that are the possible distributions on the source sequence. We then consider algorithms for encoding the source sequence given a known or "estimated" distribution on the sequence. In particular, we describe arithmetic coding, which is an extension of the Shannon–Fano–Elias code of Section 5.9 that permits incremental encoding and decoding of sequences of source symbols.

We then describe two basic versions of the class of adaptive dictionary compression algorithms called Lempel–Ziv, based on the papers by Ziv and Lempel [603, 604]. We provide a proof of asymptotic optimality for these algorithms, showing that in the limit they achieve the entropy rate for any stationary ergodic source. In Chapter 16 we extend the notion of universality to investment in the stock market and describe online portfolio selection procedures that are analogous to the universal methods for data compression.

13.1 UNIVERSAL CODES AND CHANNEL CAPACITY

Assume that we have a random variable X drawn according to a distribution from the family $\{p_\theta\}$, where the parameter $\theta \in \{1, 2, \ldots, m\}$ is unknown. We wish to find an efficient code for this source.

From the results of Chapter 5, if we know θ, we can construct a code with codeword lengths $l(x) = \log \frac{1}{p_\theta(x)}$, achieving an average codeword

length equal to the entropy $H_\theta(x) = -\sum_x p_\theta(x) \log p_\theta(x)$, and this is the best that we can do. For the purposes of this section, we will ignore the integer constraints on $l(x)$, knowing that applying the integer constraint will cost at most one bit in expected length. Thus,

$$\min_{l(x)} E_{p_\theta}[l(X)] = E_{p_\theta}\left[\log \frac{1}{p_\theta(X)}\right] = H(p_\theta). \tag{13.1}$$

What happens if we do not know the true distribution p_θ, yet wish to code as efficiently as possible? In this case, using a code with codeword lengths $l(x)$ and implied probability $q(x) = 2^{-l(x)}$, we define the redundancy of the code as the difference between the expected length of the code and the lower limit for the expected length:

$$R(p_\theta, q) = E_{p_\theta}[l(X)] - E_{p_\theta}\left[\log \frac{1}{p_\theta(X)}\right] \tag{13.2}$$

$$= \sum_x p_\theta(x) \left(l(x) - \log \frac{1}{p(x)}\right) \tag{13.3}$$

$$= \sum_x p_\theta(x) \left(\log \frac{1}{q(x)} - \log \frac{1}{p(x)}\right) \tag{13.4}$$

$$= \sum_x p_\theta(x) \log \frac{p_\theta(x)}{q(x)} \tag{13.5}$$

$$= D(p_\theta \| q), \tag{13.6}$$

where $q(x) = 2^{-l(x)}$ is the distribution that corresponds to the codeword lengths $l(x)$.

We wish to find a code that does well irrespective of the true distribution p_θ, and thus we define the *minimax redundancy* as

$$R^* = \min_q \max_{p_\theta} R(p_\theta, q) = \min_q \max_{p_\theta} D(p_\theta \| q). \tag{13.7}$$

This minimax redundancy is achieved by a distribution q that is at the "center" of the information ball containing the distributions p_θ, that is, the distribution q whose maximum distance from any of the distributions p_θ is minimized (Figure 13.1).

To find the distribution q that is as close as possible to all the possible p_θ in relative entropy, consider the following channel:

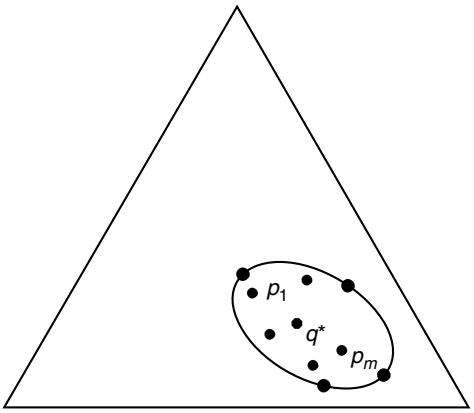

FIGURE 13.1. Minimum radius information ball containing all the p_θ's

$$\theta \to \begin{bmatrix} \cdots p_1 \cdots \\ \cdots p_2 \cdots \\ \vdots \\ \cdots p_\theta \cdots \\ \vdots \\ \cdots p_m \cdots \end{bmatrix} \to X. \tag{13.8}$$

This is a channel $\{\theta, p_\theta(x), \mathcal{X}\}$ with the rows of the transition matrix equal to the different p_θ's, the possible distributions of the source. We will show that the minimax redundancy R^* is equal to the capacity of this channel, and the corresponding optimal coding distribution is the output distribution of this channel induced by the capacity-achieving input distribution. The capacity of this channel is given by

$$C = \max_{\pi(\theta)} I(\theta; X) = \max_{\pi(\theta)} \sum_\theta \pi(\theta) p_\theta(x) \log \frac{p_\theta(x)}{q_\pi(x)}, \tag{13.9}$$

where

$$q_\pi(x) = \sum_\theta \pi(\theta) p_\theta(x). \tag{13.10}$$

The equivalence of R^* and C is expressed in the following theorem:

Theorem 13.1.1 *(Gallager [229], Ryabko [450]) The capacity of a channel $p(x|\theta)$ with rows $p_1, p_2, \ldots, p_m$ is given by*

$$C = R^* = \min_q \max_\theta D(p_\theta \| q). \tag{13.11}$$

The distribution q that achieves the minimum in (13.11) is the output distribution $q^*(x)$ induced be the capacity-achieving input distribution $\pi^*(\theta)$:

$$q^*(x) = q_{\pi^*}(x) = \sum_\theta \pi^*(\theta) p_\theta(x). \tag{13.12}$$

Proof: Let $\pi(\theta)$ be an input distribution on $\theta \in \{1, 2, \ldots, m\}$, and let the induced output distribution be q_π:

$$(q_\pi)_j = \sum_{i=1}^{m} \pi_i p_{ij}, \tag{13.13}$$

where $p_{ij} = p_\theta(x)$ for $\theta = i$, $x = j$. Then for any distribution q on the output, we have

$$I_\pi(\theta; X) = \sum_{i,j} \pi_i p_{ij} \log \frac{p_{ij}}{(q_\pi)_j} \tag{13.14}$$

$$= \sum_i \pi_i D(p_i \| q_\pi) \tag{13.15}$$

$$= \sum_{i,j} \pi_i p_{ij} \log \frac{p_{ij}}{q_j} \frac{q_j}{(q_\pi)_j} \tag{13.16}$$

$$= \sum_{i,j} \pi_i p_{ij} \log \frac{p_{ij}}{q_j} + \sum_{i,j} \pi_i p_{ij} \log \frac{q_j}{(q_\pi)_j} \tag{13.17}$$

$$= \sum_{i,j} \pi_i p_{ij} \log \frac{p_{ij}}{q_j} + \sum_j (q_\pi)_j \log \frac{q_j}{(q_\pi)_j} \tag{13.18}$$

$$= \sum_{i,j} \pi_i p_{ij} \log \frac{p_{ij}}{q_j} - D(q_\pi \| q) \tag{13.19}$$

$$= \sum_i \pi_i D(p_i \| q) - D(q_\pi \| q) \tag{13.20}$$

$$\le \sum_i \pi_i D(p_i \| q) \tag{13.21}$$

for all q, with equality iff $q = q_\pi$. Thus, for all q,

$$\sum_i \pi_i D(p_i \| q) \ge \sum_i \pi_i D(p_i \| q_\pi), \tag{13.22}$$

and therefore

$$I_\pi(\theta; X) = \min_q \sum_i \pi_i D(p_i \| q) \tag{13.23}$$

is achieved when $q = q_\pi$. Thus, the output distribution that minimizes the average distance to all the rows of the transition matrix is the the output distribution induced by the channel (Lemma 10.8.1).

The channel capacity can now be written as

$$C = \max_\pi I_\pi(\theta; X) \tag{13.24}$$

$$= \max_\pi \min_q \sum_i \pi_i D(p_i \| q). \tag{13.25}$$

We can now apply a fundamental theorem of game theory, which states that for a continuous function $f(x, y)$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$, if $f(x, y)$ is convex in x and concave in y, and $\mathcal{X}$, $\mathcal{Y}$ are compact convex sets, then

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y) = \max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} f(x, y). \tag{13.26}$$

The proof of this minimax theorem can be found in [305, 392].

By convexity of relative entropy (Theorem 2.7.2), $\sum_i \pi_i D(p_i \| q)$ is convex in q and concave in π, and therefore

$$C = \max_\pi \min_q \sum_i \pi_i D(p_i \| q) \tag{13.27}$$

$$= \min_q \max_\pi \sum_i \pi_i D(p_i \| q) \tag{13.28}$$

$$= \min_q \max_i D(p_i \| q), \tag{13.29}$$

where the last equality follows from the fact that the maximum is achieved by putting all the weight on the index i maximizing $D(p_i \| q)$ in (13.28). It also follows that $q^* = q_{\pi^*}$. This completes the proof. □

Thus, the channel capacity of the channel from θ to X is the minimax expected redundancy in source coding.

Example 13.1.1 Consider the case when $\mathcal{X} = \{1, 2, 3\}$ and θ takes only two values, 1 and 2, and the corresponding distributions are $p_1 = (1 - \alpha, \alpha, 0)$ and $p_2 = (0, \alpha, 1 - \alpha)$. We would like to encode a sequence of symbols from $\mathcal{X}$ without knowing whether the distribution is p_1 or p_2. The arguments above indicate that the worst-case optimal code uses the

codeword lengths corresponding to the distribution that has a minimal relative entropy distance from both distributions, in this case, the midpoint of the two distributions. Using this distribution, $q = \left\{\frac{1-\alpha}{2}, \alpha, \frac{1-\alpha}{2}\right\}$, we achieve a redundancy of

$$D(p_1\|q) = D(p_2\|q) = (1-\alpha)\log\frac{1-\alpha}{(1-\alpha)/2} + \alpha\log\frac{\alpha}{\alpha} + 0 = 1-\alpha. \tag{13.30}$$

The channel with transition matrix rows equal to p_1 and p_2 is equivalent to the erasure channel (Section 7.1.5), and the capacity of this channel can easily be calculated to be $(1-\alpha)$, achieved with a uniform distribution on the inputs. The output distribution corresponding to the capacity-achieving input distribution is equal to $\left\{\frac{1-\alpha}{2}, \alpha, \frac{1-\alpha}{2}\right\}$ (i.e., the same as the distribution q above). Thus, if we don't know the distribution for this class of sources, we code using the distribution q rather than p_1 or p_2, and incur an additional cost of $1-\alpha$ bits per source symbol above the ideal entropy bound.

13.2 UNIVERSAL CODING FOR BINARY SEQUENCES

Now we consider an important special case of encoding a binary sequence $x^n \in \{0,1\}^n$. We do not make any assumptions about the probability distribution for $x_1, x_2, \ldots, x_n$.

We begin with bounds on the size of $\binom{n}{k}$, taken from Wozencraft and Reiffen [567] proved in Lemma 17.5.1: For $k \neq 0$ or n,

$$\sqrt{\frac{n}{8k(n-k)}} \leq \binom{n}{k} 2^{-nH(k/n)} \leq \sqrt{\frac{n}{\pi k(n-k)}}. \tag{13.31}$$

We first describe an offline algorithm to describe the sequence; we count the number of 1's in the sequence, and after we have seen the entire sequence, we send a two-stage description of the sequence. The first stage is a count of the number of 1's in the sequence [i.e., $k = \sum_i x_i$ (using $\lceil \log(n+1) \rceil$ bits)], and the second stage is the index of this sequence among all sequences that have k 1's (using $\lceil \log \binom{n}{k} \rceil$ bits). This two-stage description requires total length

$$l(x^n) \leq \log(n+1) + \log\binom{n}{k} + 2 \tag{13.32}$$

$$\leq \log n + nH\left(\frac{k}{n}\right) - \frac{1}{2}\log n - \frac{1}{2}\log\left(\pi\frac{k}{n}\frac{(n-k)}{n}\right) + 3 \tag{13.33}$$

$$= nH\left(\frac{k}{n}\right) + \frac{1}{2}\log n - \frac{1}{2}\log\left(\pi\frac{k}{n}\frac{n-k}{n}\right) + 3. \tag{13.34}$$

Thus, the cost of describing the sequence is approximately $\frac{1}{2}\log n$ bits above the optimal cost with the Shannon code for a Bernoulli distribution corresponding to k/n. The last term is unbounded at $k = 0$ or $k = n$, so the bound is not useful for these cases (the actual description length is $\log(n+1)$ bits, whereas the entropy $H(k/n) = 0$ when $k = 0$ or $k = n$).

This counting approach requires the compressor to wait until he has seen the entire sequence. We now describe a different approach using a mixture distribution that achieves the same result on the fly. We choose the coding distribution $q(x_1, x_2, \ldots, x_n) = 2^{-l(x_1,x_2,\ldots,x_n)}$ to be a uniform mixture of all Bernoulli(θ) distributions on $x_1, x_2, \ldots, x_n$. We will analyze the performance of a code using this distribution and show that such codes perform well for all input sequences.

We construct this distribution by assuming that θ, the parameter of the Bernoulli distribution is drawn according to a uniform distribution on [0, 1]. The probability of a sequence $x_1, x_2, \ldots, x_n$ with k ones is $\theta^k(1-\theta)^{n-k}$ under the Bernoulli(θ) distribution. Thus, the mixture probability of the sequence is

$$p(x_1, x_2, \ldots, x_n) = \int_0^1 \theta^k(1-\theta)^{n-k}d\theta \triangleq A(n,k). \tag{13.35}$$

Integrating by parts, setting $u = (1-\theta)^{n-k}$ and $dv = \theta^k d\theta$, we have

$$\int_0^1 \theta^k(1-\theta)^{n-k}d\theta = \left[\frac{1}{k+1}\theta^{k+1}(1-\theta)^{n-k}\right]_0^1 + \frac{n-k}{k+1}\int_0^1 \theta^{k+1}(1-\theta)^{n-k-1}d\theta, \tag{13.36}$$

or

$$A(n,k) = \frac{n-k}{k+1}A(n,k+1). \tag{13.37}$$

Now $A(n,n) = \int_0^1 \theta^n d\theta = \frac{1}{n+1}$, and we can easily verify from the recursion that

$$p(x_1, x_2, \ldots, x_n) = A(n,k) = \frac{1}{n+1}\frac{1}{\binom{n}{k}}. \tag{13.38}$$

The codeword length with respect to the mixture distribution is

$$\left\lceil \log \frac{1}{q(x^n)} \right\rceil \leq \log(n+1) + \log \binom{n}{k} + 1, \tag{13.39}$$

which is within one bit of the length of the two-stage description above. Thus, we have a similar bound on the codeword length

$$l(x_1, x_2, \ldots, x_n) \leq H\left(\frac{k}{n}\right) + \frac{1}{2}\log n - \frac{1}{2}\log\left(\pi \frac{k}{n}\frac{(n-k)}{n}\right) + 2 \tag{13.40}$$

for all sequences $x_1, x_2, \ldots, x_n$. This mixture distribution achieves a codeword length within $\frac{1}{2}\log n$ bits of the optimal code length $nH(k/n)$ that would be required if the source were really Bernoulli(k/n), without any assumptions about the distribution of the source.

This mixture distribution yields a nice expression for the conditional probability of the next symbol given the previous symbols of $x_1, x_2, \ldots, x_n$. Let k_i be the number of 1's in the first i symbols of $x_1, x_2, \ldots, x_n$. Using (13.38), we have

$$q(x_{i+1} = 1 | x^i) = \frac{q(x^i, 1)}{q(x^i)} \tag{13.41}$$

$$= \left(\frac{1}{i+2}\frac{1}{\binom{i+1}{k_i+1}}\right) \Big/ \left(\frac{1}{i+1}\frac{1}{\binom{i}{k_i}}\right) \tag{13.42}$$

$$= \frac{1}{i+2}\frac{(k_i+1)!(n-k_i)!}{(i+1)!}(i+1)\frac{k_i!(i-k_i)!}{i!} \tag{13.43}$$

$$= \frac{k_i+1}{i+2}. \tag{13.44}$$

This is the Bayesian posterior probability of 1 given the uniform prior on θ, and is called the *Laplace estimate* for the probability of the next symbol. We can use this posterior probability as the probability of the next symbol for arithmetic coding, and achieve the codeword length $\log \frac{1}{q(x^n)}$ in a sequential manner with finite-precision arithmetic. This is a horizon-free result, in that the procedure does not depend on the length of the sequence.

One issue with the uniform mixture approach or the two-stage approach is that the bound does not apply for $k = 0$ or $k = n$. The only uniform bound that we can give on the extra redundancy is $\log n$, which we can obtain by using the bounds of (11.40). The problem is that

we are not assigning enough probability to sequences with $k = 0$ or $k = n$. If instead of using a uniform distribution on θ, we used the Dirichlet$(\frac{1}{2}, \frac{1}{2})$distribution, also called the Beta$(\frac{1}{2}, \frac{1}{2})$ distribution, the probability of a sequence $x_1, x_2, \ldots, x_n$ becomes

$$q_{\frac{1}{2}}(x^n) = \int_0^1 \theta^k (1-\theta)^{n-k} \frac{1}{\pi\sqrt{\theta(1-\theta)}}\, d\theta \tag{13.45}$$

and it can be shown that this achieves a description length

$$\log \frac{1}{q_{\frac{1}{2}}(x^n)} \leq H(k/n) + \frac{1}{2}\log n + \log \frac{\pi}{8} \tag{13.46}$$

for all $x^n \in \{0, 1\}^n$, achieving a uniform bound on the redundancy of the universal mixture code. As in the case of the uniform prior, we can calculate the conditional distribution of the next symbol, given the previous observations, as

$$q_{\frac{1}{2}}(x_{i+1} = 1 | x^i) = \frac{k_i + \frac{1}{2}}{i+1}, \tag{13.47}$$

which can be used with arithmetic coding to provide an online algorithm to encode the sequence. We will analyze the performance of the mixture algorithm in greater detail when we analyze universal portfolios in Section 16.7.

13.3 ARITHMETIC CODING

The Huffman coding procedure described in Chapter 5 is optimal for encoding a random variable with a known distribution that has to be encoded symbol by symbol. However, due to the fact that the codeword lengths for a Huffman code were restricted to be integral, there could be a loss of up to 1 bit per symbol in coding efficiency. We could alleviate this loss by using blocks of input symbols—however, the complexity of this approach increases exponentially with block length. We now describe a method of encoding without this inefficiency. In arithmetic coding, instead of using a sequence of bits to represent a symbol, we represent it by a subinterval of the unit interval.

The code for a sequence of symbols is an interval whose length decreases as we add more symbols to the sequence. This property allows us to have a coding scheme that is incremental (the code for an extension to a sequence can be calculated simply from the code for the original sequence) and for which the codeword lengths are not restricted to be integral. The motivation

for arithmetic coding is based on Shannon–Fano–Elias coding (Section 5.9) and the following lemma:

Lemma 13.3.1 *Let Y be a random variable with continuous probability distribution function $F(y)$. Let $U = F(Y)$ (i.e., U is a function of Y defined by its distribution function). Then U is uniformly distributed on* $[0, 1]$.

Proof: Since $F(y) \in [0, 1]$, the range of U is $[0, 1]$. Also, for $u \in [0, 1]$,

$$F_U(u) = \Pr(U \le u) \tag{13.48}$$
$$= \Pr(F(Y) \le u) \tag{13.49}$$
$$= \Pr(Y \le F^{-1}(u)) \tag{13.50}$$
$$= F(F^{-1}(u)) \tag{13.51}$$
$$= u, \tag{13.52}$$

which proves that U has a uniform distribution in $[0, 1]$. □

Now consider an infinite sequence of random variables $X_1, X_2, \ldots$ from a finite alphabet $\mathcal{X} = 0, 1, 2, \ldots, m$. For any sequence $x_1, x_2, \ldots,$ from this alphabet, we can place 0. in front of the sequence and consider it as a real number (base $m + 1$) between 0 and 1. Let X be the real-valued random variable $X = 0.X_1X_2\ldots$. Then X has the following distribution function:

$$F_X(x) = \Pr\{X \le x = 0.x_1x_2\cdots\} \tag{13.53}$$
$$= \Pr\{0.X_1X_2\cdots \le 0.x_1x_2\cdots\} \tag{13.54}$$
$$= \Pr\{X_1 < x_1\} + \Pr\{X_1 = x_1, X_2 < x_2\} + \cdots. \tag{13.55}$$

Now let $U = F_X(X) = F_X(0.X_1X_2\ldots) = 0.F_1F_2\ldots$. If the distribution on infinite sequences X^∞ has no atoms, then, by the lemma above, U has a uniform distribution on $[0, 1]$, and therefore the bits $F_1F_2\ldots$ in the binary expansion of U are Bernoulli($\frac{1}{2}$) (i.e., they are independent and uniformly distributed on $\{0, 1\}$). These bits are therefore incompressible, and form a compressed representation of the sequence $0.X_1X_2\ldots$. For Bernoulli or Markov models, it is easy to calculate the cumulative distribution function, as illustrated in the following example.

Example 13.3.1 Let $X_1, X_2, \ldots, X_n$ be Bernoulli(p). Then the sequence $x^n = 110101$ maps into

$$\begin{aligned} F(x^n) &= \Pr(X_1 < 1) + \Pr(X_1 = 1, X_2 < 1) \\ &\quad + \Pr(X_1 = 1, X_2 = 1, X_3 < 0) \\ &\quad + \Pr(X_1 = 1, X_2 = 1, X_3 = 0, X_4 < 1) \\ &\quad + \Pr(X_1 = 1, X_2 = 1, X_3 = 0, X_4 = 1, X_5 < 0) \\ &\quad + \Pr(X_1 = 1, X_2 = 1, X_3 = 0, X_4 = 1, X_5 = 0, X_6 < 1) \end{aligned} \tag{13.56}$$

$$= q + pq + p^2 \cdot 0 + p^2 q \cdot q + p^2 qp \cdot 0 + p^2 qpqq \tag{13.57}$$

$$= q + pq + p^2 q^2 + p^3 q^3. \tag{13.58}$$

Note that each term is easily computed from the previous terms. In general, for an arbitrary binary process $\{X_i\}$,

$$F(x^n) = \sum_{k=1}^{n} p(x^{k-1}0)x_k. \tag{13.59}$$

The probability transform thus forms an invertible mapping from infinite source sequences to incompressible infinite binary sequences. We now consider the compression achieved by this transformation on finite sequences. Let $X_1, X_2, \ldots, X_n$ be a sequence of binary random variables of length n, and let $x_1, x_2, \ldots, x_n$ be a particular outcome. We can treat this sequence as representing an interval $[0.x_1x_2\ldots x_n000\ldots, 0.x_1x_2\ldots x_n1111\ldots)$, or equivalently, $[0.x_1x_2\ldots x_n, 0.x_1x_2\ldots x_n + (\frac{1}{2})^n)$. This is the set of infinite sequences that start with $0.x_1x_2\cdots x_n$. Under the probability transform, this interval gets mapped into another interval, $[F_Y(0.x_1x_2\cdots x_n), F_Y(0.x_1x_2\cdots x_n + (\frac{1}{2})^n))$, whose length is equal to $P_X(x_1, x_2, \ldots, x_n)$, the sum of the probabilities of all infinite sequences that start with $0.x_1x_2\cdots x_n$. Under the probability inverse transform, any real number u within this interval maps into a sequence that starts with $x_1, x_2, \ldots, x_n$, and therefore given u and n, we can reconstruct $x_1, x_2, \ldots, x_n$. The Shannon–Fano–Elias coding scheme described earlier allows one to construct a prefix-free code of length $\log \frac{1}{p(x_1,x_2,\ldots,x_n)} + 2$ bits, and therefore it is possible to encode the sequence $x_1, x_2, \ldots, x_n$ with this length. Note that $\log \frac{1}{p(x_1,\ldots,x_n)}$ is the ideal codeword length for x^n.

The process of encoding the sequence with the cumulative distribution function described above assumes arbitrary accuracy for the computation. In practice, though, we have to implement all numbers with finite precision, and we describe such an implementation. The key is to consider

not infinite-precision points for the cumulative distribution function but intervals in the unit interval. Any finite-length sequence of symbols can be said to correspond to a subinterval of the unit interval. The objective of the arithmetic coding algorithm is to represent a sequence of random variables by a subinterval in [0, 1]. As the algorithm observes more input symbols, the length of the subinterval corresponding to the input sequence decreases. As the top end of the interval and the bottom end of the interval get closer, they begin to agree in the first few bits. These will be first few bits of the output sequence. As soon as the two ends of the interval agree, we can output the corresponding bits. We can therefore shift these bits out of the calculation and effectively scale the remaining intervals so that entire calculation can be done with finite precision. We will not go into the details here—there is a very good description of the algorithm and performance considerations in Bell et al. [41]

Example 13.3.2 (*Arithmetic coding for a ternary input alphabet*) Consider a random variable X with a ternary alphabet $\{A, B, C\}$, which are assumed to have probabilities 0.4, 0.4, and 0.2, respectively. Let the sequence to be encoded by ACAA. Thus, $F_l(\cdot) = (0, 0.4, 0.8)$ and $F_h(\cdot) = (0.4, 0.8, 1.0)$. Initially, the input sequence is empty, and the corresponding interval is [0, 1). The cumulative distribution function after the first input symbol is shown in Figure 13.2. It is easy to calculate that the interval in the algorithm without scaling after the first symbol A is

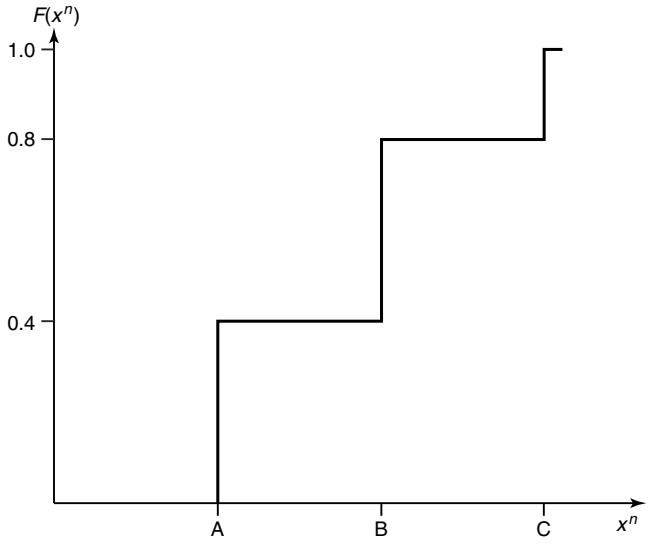

FIGURE 13.2. Cumulative distribution function after the first symbol.

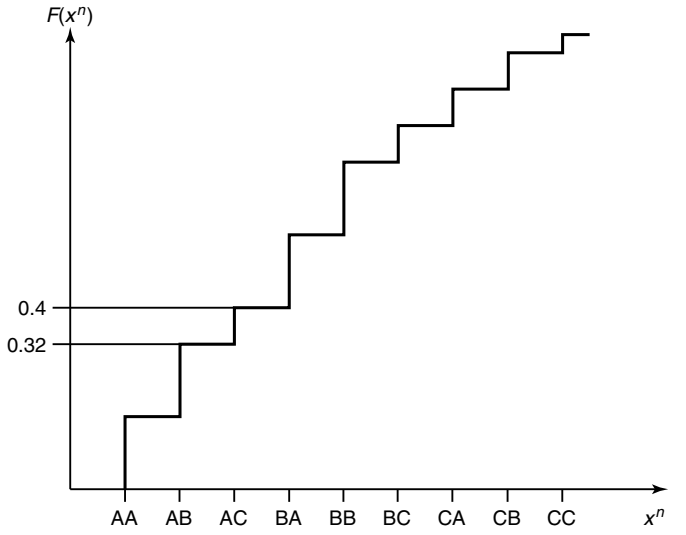

FIGURE 13.3. Cumulative distribution function after the second symbol.

[0, 0.4); after the second symbol, C, it is [0.32, 0.4) (Figure 13.3); after the third symbol A, it is [0.32,0.352); and after the fourth symbol A, it is [0.32, 0.3328). Since the probability of this sequence is 0.0128, we will use $\log(1/0.0128) + 2$ (i.e., 9 bits) to encode the midpoint of the interval sequence using Shannon–Fano–Elias coding (0.3264, which is 0.010100111 binary).

In summary, the arithmetic coding procedure, given any length n and probability mass function $q(x_1x_2 \cdots x_n)$, enables one to encode the sequence $x_1x_2 \cdots x_n$ in a code of length $\log \frac{1}{q(x_1x_2 \cdots x_n)} + 2$ bits. If the source is i.i.d. and the assumed distribution q is equal to the true distribution p of the data, this procedure achieves an average length for the block that is within 2 bits of the entropy. Although this is not necessarily optimal for any fixed block length (a Huffman code designed for the distribution could have a lower average codeword length), the procedure is incremental and can be used for any blocklength.

13.4 LEMPEL–ZIV CODING

In Section 13.3 we discussed the basic ideas of arithmetic coding and mentioned some results on worst-case redundancy for coding a sequence from an unknown distribution. We now discuss a popular class of techniques for source coding that are universally optimal (their asymptotic

compression rate approaches the entropy rate of the source for any stationary ergodic source) and simple to implement. This class of algorithms is termed Lempel–Ziv, named after the authors of two seminal papers [603, 604] that describe the two basic algorithms that underlie this class. The algorithms could also be described as adaptive dictionary compression algorithms.

The notion of using dictionaries for compression dates back to the invention of the telegraph. At the time, companies were charged by the number of letters used, and many large companies produced codebooks for the frequently used phrases and used the codewords for their telegraphic communication. Another example is the notion of greetings telegrams that are popular in India—there is a set of standard greetings such as "25:Merry Christmas" and "26:May Heaven's choicest blessings be showered on the newly married couple." A person wishing to send a greeting only needs to specify the number, which is used to generate the actual greeting at the destination.

The idea of adaptive dictionary-based schemes was not explored until Ziv and Lempel wrote their papers in 1977 and 1978. The two papers describe two distinct versions of the algorithm. We refer to these versions as LZ77 or sliding window Lempel–Ziv and LZ78 or tree-structured Lempel–Ziv. (They are sometimes called LZ1 and LZ2, respectively.)

We first describe the basic algorithms in the two cases and describe some simple variations. We later prove their optimality, and end with some practical issues. The key idea of the Lempel–Ziv algorithm is to parse the string into phrases and to replace phrases by pointers to where the same string has occurred in the past. The differences between the algorithms is based on differences in the set of possible match locations (and match lengths) the algorithm allows.

13.4.1 Sliding Window Lempel–Ziv Algorithm

The algorithm described in the 1977 paper encodes a string by finding the longest match anywhere within a window of past symbols and represents the string by a pointer to location of the match within the window and the length of the match. There are many variations of this basic algorithm, and we describe one due to Storer and Szymanski [507].

We assume that we have a string $x_1, x_2, \ldots$ to be compressed from a finite alphabet. A *parsing* S of a string $x_1 x_2 \cdots x_n$ is a division of the string into phrases, separated by commas. Let W be the length of the window. Then the algorithm can be described as follows: Assume that we have compressed the string until time $i-1$. Then to find the next phrase, find the largest k such that for some j, $i-1-W \leq j \leq i-1$,

the string of length k starting at x_j is equal to the string (of length k) starting at x_i (i.e., $x_{j+l} = x_{i+l}$ for all $0 \leq l < k$). The next phrase is then of length k (i.e., $x_i \ldots x_{i+k-1}$) and is represented by the pair (P, L), where P is the location of the beginning of the match and L is the length of the match. If a match is not found in the window, the next character is sent uncompressed. To distinguish between these two cases, a flag bit is needed, and hence the phrases are of two types: (F, P, L) or (F, C), where C represents an uncompressed character.

Note that the target of a (pointer,length) pair could extend beyond the window, so that it overlaps with the new phrase. In theory, this match could be arbitrarily long; in practice, though, the maximum phrase length is restricted to be less than some parameter.

For example, if $W = 4$ and the string is ABBABBABBBAABABA and the initial window is empty, the string will be parsed as follows: A,B,B,ABBABB,BA,A,BA,BA, which is represented by the sequence of "pointers": (0,A),(0,B),(1,1,1),(1,3,6),(1,4,2),(1,1,1),(1,3,2),(1,2,2), where the flag bit is 0 if there is no match and 1 if there is a match, and the location of the match is measured backward from the end of the window. [In the example, we have represented every match within the window using the (P, L) pair; however, it might be more efficient to represent short matches as uncompressed characters. See Problem 13.8 for details.]

We can view this algorithm as using a dictionary that consists of all substrings of the string in the window and of all single characters. The algorithm finds the longest match within the dictionary and sends a pointer to that match. We later show that a simple variation on this version of LZ77 is asymptotically optimal. Most practical implementations of LZ77, such as gzip and pkzip, are also based on this version of LZ77.

13.4.2 Tree-Structured Lempel–Ziv Algorithms

In the 1978 paper, Ziv and Lempel described an algorithm that parses a string into phrases, where each phrase is the shortest phrase not seen earlier. This algorithm can be viewed as building a dictionary in the form of a tree, where the nodes correspond to phrases seen so far. The algorithm is particularly simple to implement and has become popular as one of the early standard algorithms for file compression on computers because of its speed and efficiency. It is also used for data compression in high-speed modems.

The source sequence is sequentially parsed into strings that have not appeared so far. For example, if the string is ABBABBABBBAABABAA ..., we parse it as A,B,BA,BB,AB,BBA,ABA,BAA.... After every comma, we look along the input sequence until we come to the shortest string that has not been marked off before. Since this is the shortest such string,

all its prefixes must have occurred earlier. (Thus, we can build up a tree of these phrases.) In particular, the string consisting of all but the last bit of this string must have occurred earlier. We code this phrase by giving the location of the prefix and the value of the last symbol. Thus, the string above would be represented as (0,A),(0,B),(2,A),(2,B),(1,B),(4,A),(5,A),(3,A), ...

Sending an uncompressed character in each phrase results in a loss of efficiency. It is possible to get around this by considering the extension character (the last character of the current phrase) as part of the next phrase. This variation, due to Welch [554], is the basis of most practical implementations of LZ78, such as compress on Unix, in compression in modems, and in the image files in the GIF format.

13.5 OPTIMALITY OF LEMPEL–ZIV ALGORITHMS

13.5.1 Sliding Window Lempel–Ziv Algorithms

In the original paper of Ziv and Lempel [603], the authors described the basic LZ77 algorithm and proved that it compressed any string as well as any finite-state compressor acting on that string. However, they did not prove that this algorithm achieved asymptotic optimality (i.e., that the compression ratio converged to the entropy for an ergodic source). This result was proved by Wyner and Ziv [591].

The proof relies on a simple lemma due to Kac: the average length of time that you need to wait to see a particular symbol is the reciprocal of the probability of a symbol. Thus, we are likely to see the high-probability strings within the window and encode these strings efficiently. The strings that we do not find within the window have low probability, so that asymptotically, they do not influence the compression achieved.

Instead of proving the optimality of the practical version of LZ77, we will present a simpler proof for a different version of the algorithm, which, though not practical, captures some of the basic ideas. This algorithm assumes that both the sender and receiver have access to the infinite past of the string, and represents a string of length n by pointing to the last time it occurred in the past.

We assume that we have a stationary and ergodic process defined for time from $-\infty$ to ∞, and that both the encoder and decoder have access to $\ldots, X_{-2}, X_{-1}$, the infinite past of the sequence. Then to encode $X_0, X_1, \ldots, X_{n-1}$ (a block of length n), we find the last time we have seen these n symbols in the past. Let

$$R_n(X_0, X_1, \ldots, X_{n-1}) = \max\{j < 0 : (X_{-j}, X_{-j+1} \ldots X_{-j+n-1}) = (X_0, \ldots, X_{n-1})\}. \quad (13.60)$$

Then to represent $X_0, \ldots, X_{n-1}$, we need only to send R_n to the receiver, who can then look back R_n bits into the past and recover $X_0, \ldots, X_{n-1}$. Thus, the cost of the encoding is the cost of representing R_n. We will show that this cost is approximately $\log R_n$ and that asymptotically $\frac{1}{n} E \log R_n \rightarrow H(\mathcal{X})$, thus proving the asymptotic optimality of this algorithm.

We will need the following lemmas.

Lemma 13.5.1 *There exists a prefix-free code for the integers such that the length of the codeword for integer k is $\log k + 2 \log \log k + O(1)$.*

Proof: If we knew that $k \leq m$, we could encode k with $\log m$ bits. However, since we don't have an upper limit for k, we need to tell the receiver the length of the encoding of k (i.e., we need to specify $\log k$). Consider the following encoding for the integer k: We first represent $\lceil \log k \rceil$ in unary, followed by the binary representation of k:

$$C_1(k) = \underbrace{00\cdots0}_{\lceil \log k \rceil \text{ 0's}} \; 1 \; \underbrace{xx\cdots x}_{k \text{ in binary}} \quad . \tag{13.61}$$

It is easy to see that the length of this representation is $2\lceil \log k \rceil + 1 \leq 2 \log k + 3$. This is more than the length we are looking for since we are using the very inefficient unary code to send $\log k$. However, if we use C_1 to represent $\log k$, it is now easy to see that this representation has a length less than $\log k + 2 \log \log k + 4$, which proves the lemma. A similar method is presented in the discussion following Theorem 14.2.3. □

The key result that underlies the proof of the optimality of LZ77 is Kac's lemma, which relates the average recurrence time to the probability of a symbol for any stationary ergodic process. For example, if $X_1, X_2, \ldots, X_n$ is an i.i.d. process, we ask what is the expected waiting time to see the symbol a again, conditioned on the fact that $X_1 = a$. In this case, the waiting time has a geometric distribution with parameter $p = p(X_0 = a)$, and thus the expected waiting time is $1/p(X_0 = a)$. The somewhat surprising result is that the same is true even if the process is not i.i.d., but stationary and ergodic. A simple intuitive reason for this is that in a long sample of length n, we would expect to see a about $np(a)$ times, and the average distance between these occurrences of a is $n/(np(a))$ (i.e., $1/p(a)$).

Lemma 13.5.2 (*Kac*) *Let $\ldots, U_2, U_1, U_0, U_1, \ldots$ be a stationary ergodic process on a countable alphabet. For any u such that $p(u) > 0$*

and for $i = 1, 2, \ldots,$ *let*

$$Q_u(i) = \Pr\left\{U_{-i} = u; U_j \neq u \text{ for } -i < j < 0 | U_0 = u\right\} \tag{13.62}$$

[i.e., $Q_u(i)$ *is the conditional probability that the most recent previous occurrence of the symbol* u *is* i*, given that* $U_0 = u$*]. Then*

$$E(R_1(U)|X_0 = u) = \sum_i i\, Q_u(i) = \frac{1}{p(u)}. \tag{13.63}$$

Thus, the conditional expected waiting time to see the symbol u *again, looking backward from zero, is* $1/p(u)$*.*

Note the amusing fact that the expected recurrence time

$$E R_1(U) = \sum p(u)\frac{1}{p(u)} = m, \tag{13.64}$$

where m is the alphabet size.

Proof: Let $U_0 = u$. Define the events for $j = 1, 2, \ldots$ and $k = 0, 1, 2, \ldots$:

$$A_{jk} = \left\{U_{-j} = u,\ U_l \neq u, -j < l < k,\ U_k = u\right\}. \tag{13.65}$$

Event A_{jk} corresponds to the event where the last time before zero at which the process is equal to u is at $-j$, the first time after zero at which the process equals u is k. These events are disjoint, and by ergodicity, the probability $\Pr\{\cup_{j,k} A_{jk}\} = 1$. Thus,

$$1 = \Pr\left\{\cup_{j,k} A_{jk}\right\} \tag{13.66}$$

$$\stackrel{(a)}{=} \sum_{j=1}^{\infty}\sum_{k=0}^{\infty} \Pr\{A_{jk}\} \tag{13.67}$$

$$= \sum_{j=1}^{\infty}\sum_{k=0}^{\infty} \Pr(U_k = u) \Pr\left\{U_{-j} = u, U_l \neq u, -j < l < k | U_k = u\right\} \tag{13.68}$$

$$\stackrel{(b)}{=} \sum_{j=1}^{\infty}\sum_{k=0}^{\infty} \Pr(U_k = u) Q_u(j + k) \tag{13.69}$$

$$\stackrel{(c)}{=} \sum_{j=1}^{\infty}\sum_{k=0}^{\infty} \Pr(U_0 = u) Q_u(j+k) \tag{13.70}$$

$$= \Pr(U_0 = u) \sum_{j=1}^{\infty}\sum_{k=0}^{\infty} Q_u(j+k) \tag{13.71}$$

$$\stackrel{(d)}{=} \Pr(U_0 = u) \sum_{i=1}^{\infty} i\, Q_u(i), \tag{13.72}$$

where (a) follows from the fact that the A_{jk} are disjoint, (b) follows from the definition of $Q_u(\cdot)$, (c) follows from stationarity, and (d) follows from the fact that there are i pairs (j, k) such that $j + k = i$ in the sum. Kac's lemma follows directly from this equation. □

Corollary *Let* $\ldots, X_{-1}, X_0, X_1, \ldots$ *be a stationary ergodic process and let* $R_n(X_0, \ldots, X_{n-1})$ *be the recurrence time looking backward as defined in (13.60). Then*

$$E\Big[R_n(X_0, \ldots, X_{n-1}) | (X_0, \ldots, X_{n-1}) = x_0^{n-1}\Big] = \frac{1}{p(x_0^{n-1})}. \tag{13.73}$$

Proof: Define a new process with $U_i = (X_i, X_{i+1}, \ldots, X_{i+n-1})$. The U process is also stationary and ergodic, and thus by Kac's lemma the average recurrence time for U conditioned on $U_0 = u$ is $1/p(u)$. Translating this to the X process proves the corollary. □

We are now in a position to prove the main result, which shows that the compression ratio for the simple version of Lempel–Ziv using recurrence time approaches the entropy. The algorithm describes X_0^{n-1} by describing $R_n(X_0^{n-1})$, which by Lemma 13.5.1 can be done with $\log R_n + 2 \log\log R_n + 4$ bits. We now prove the following theorem.

Theorem 13.5.1 *Let* $L_n(X_0^{n-1}) = \log R_n + 2\log\log R_n + O(1)$ *be the description length for* X_0^{n-1} *in the simple algorithm described above. Then*

$$\frac{1}{n} E L_n(X_0^{n-1}) \to H(\mathcal{X}) \tag{13.74}$$

as $n \to \infty$*, where* $H(\mathcal{X})$ *is the entropy rate of the process* $\{X_i\}$.

Proof: We will prove upper and lower bounds for EL_n. The lower bound follows directly from standard source coding results (i.e., $EL_n \geq nH$ for any prefix-free code). To prove the upper bound, we first show that

$$\overline{\lim}\frac{1}{n}E\log R_n \leq H \tag{13.75}$$

and later bound the other terms in the expression for L_n. To prove the bound for $E\log R_n$, we expand the expectation by conditioning on the value of X_0^{n-1} and then applying Jensen's inequality. Thus,

$$\frac{1}{n}E\log R_n = \frac{1}{n}\sum_{x_0^{n-1}} p(x_0^{n-1})E[\log R_n(X_0^{n-1})|X_0^{n-1} = x_0^{n-1}] \tag{13.76}$$

$$\leq \frac{1}{n}\sum_{x_0^{n-1}} p(x_0^{n-1})\log E[R_n(X_0^{n-1})|X_0^{n-1} = x_0^{n-1}] \tag{13.77}$$

$$= \frac{1}{n}\sum_{x_0^{n-1}} p(x_0^{n-1})\log \frac{1}{p(x_0^{n-1})} \tag{13.78}$$

$$= \frac{1}{n}H(X_0^{n-1}) \tag{13.79}$$

$$\searrow H(\mathcal{X}). \tag{13.80}$$

The second term in the expression for L_n is $\log\log R_n$, and we wish to show that

$$\frac{1}{n}E[\log\log R_n(X_0^{n-1})] \to 0. \tag{13.81}$$

Again, we use Jensen's inequality,

$$\frac{1}{n}E\log\log R_n \leq \frac{1}{n}\log E[\log R_n(X_0^{n-1})] \tag{13.82}$$

$$\leq \frac{1}{n}\log H(X_0^{n-1}), \tag{13.83}$$

where the last inequality follows from (13.79). For any $\epsilon > 0$, for large enough n, $H(X_0^{n-1}) < n(H+\epsilon)$, and therefore $\frac{1}{n}\log\log R_n < \frac{1}{n}\log n + \frac{1}{n}\log(H+\epsilon) \to 0$. This completes the proof of the theorem. □

Thus, a compression scheme that represents a string by encoding the last time it was seen in the past is asymptotically optimal. Of course, this scheme is not practical, since it assumes that both sender and receiver

have access to the infinite past of a sequence. For longer strings, one would have to look further and further back into the past to find a match. For example, if the entropy rate is $\frac{1}{2}$ and the string has length 200 bits, one would have to look an average of $2^{100} \approx 10^{30}$ bits into the past to find a match. Although this is not feasible, the algorithm illustrates the basic idea that matching the past is asymptotically optimal. The proof of the optimality of the practical version of LZ77 with a finite window is based on similar ideas. We will not present the details here, but refer the reader to the original proof in [591].

13.5.2 Optimality of Tree-Structured Lempel–Ziv Compression

We now consider the tree-structured version of Lempel–Ziv, where the input sequence is parsed into phrases, each phrase being the shortest string that has not been seen so far. The proof of the optimality of this algorithm has a very different flavor from the proof for LZ77; the essence of the proof is a counting argument that shows that the number of phrases cannot be too large if they are all distinct, and the probability of any sequence of symbols can be bounded by a function of the number of distinct phrases in the parsing of the sequence.

The algorithm described in Section 13.4.2 requires two passes over the string—in the first pass, we parse the string and calculate $c(n)$, the number of phrases in the parsed string. We then use that to decide how many bits $[\log c(n)]$ to allot to the pointers in the algorithm. In the second pass, we calculate the pointers and produce the coded string as indicated above. The algorithm can be modified so that it requires only one pass over the string and also uses fewer bits for the initial pointers. These modifications do not affect the asymptotic efficiency of the algorithm. Some of the implementation details are discussed by Welch [554] and Bell et al. [41].

We will show that like the sliding window version of Lempel–Ziv, this algorithm asymptotically achieves the entropy rate for the unknown ergodic source. We first define a parsing of the string to be a decomposition into phrases.

Definition A *parsing* S of a binary string $x_1x_2 \cdots x_n$ is a division of the string into phrases, separated by commas. A *distinct parsing* is a parsing such that no two phrases are identical. For example, 0,111,1 is a distinct parsing of 01111, but 0,11,11 is a parsing that is not distinct.

The LZ78 algorithm described above gives a distinct parsing of the source sequence. Let $c(n)$ denote the number of phrases in the LZ78 parsing of a sequence of length n. Of course, $c(n)$ depends on the sequence X^n. The compressed sequence (after applying the Lempel–Ziv algorithm)

consists of a list of $c(n)$ pairs of numbers, each pair consisting of a pointer to the previous occurrence of the prefix of the phrase and the last bit of the phrase. Each pointer requires $\log c(n)$ bits, and hence the total length of the compressed sequence is $c(n)[\log c(n) + 1]$ bits. We now show that $\frac{c(n)(\log c(n)+1)}{n} \to H(\mathcal{X})$ for a stationary ergodic sequence $X_1, X_2, \ldots, X_n$. Our proof is based on the simple proof of asymptotic optimality of LZ78 coding due to Wyner and Ziv [575].

Before we proceed to the details of the proof, we provide an outline of the main ideas. The first lemma shows that the number of phrases in a distinct parsing of a sequence is less than $n/\log n$; the main argument in the proof is based on the fact that there are not enough distinct short phrases. This bound holds for any distinct parsing of the sequence, not just the LZ78 parsing.

The second key idea is a bound on the probability of a sequence based on the number of distinct phrases. To illustrate this, consider an i.i.d. sequence of random variables X_1, X_2, X_3, X_4 that take on four possible values, $\{A, B, C, D\}$, with probabilities p_A, p_B, p_C, and p_D, respectively. Now consider the probability of a sequence $P(D, A, B, C) = p_D p_A p_B p_C$. Since $p_A + p_B + p_C + p_D = 1$, the product $p_D p_A p_B p_C$ is maximized when the probabilities are equal (i.e., the maximum value of the probability of a sequence of four distinct symbols is $1/256$). On the other hand, if we consider a sequence A, B, A, B, the probability of this sequence is maximized if $p_A = p_B = \frac{1}{2}$, $p_C = p_D = 0$, and the maximum probability for A, B, A, B is $\frac{1}{16}$. A sequence of the form A, A, A, A could have a probability of 1. All these examples illustrate a basic point—sequences with a large number of distinct symbols (or phrases) cannot have a large probability. Ziv's inequality (Lemma 13.5.5) is the extension of this idea to the Markov case, where the distinct symbols are the phrases of the distinct parsing of the source sequence.

Since the description length of a sequence after the parsing grows as $c \log c$, the sequences that have very few distinct phrases can be compressed efficiently and correspond to strings that could have a high probability. On the other hand, strings that have a large number of distinct phrases do not compress as well; but the probability of these sequences could not be too large by Ziv's inequality. Thus, Ziv's inequality enables us to connect the logarithm of the probability of the sequence with the number of phrases in its parsing, and this is finally used to show that the tree-structured Lempel–Ziv algorithm is asymptotically optimal.

We first prove a few lemmas that we need for the proof of the theorem. The first is a bound on the number of phrases possible in a distinct parsing of a binary sequence of length n.

Lemma 13.5.3 (*Lempel and Ziv* [604]) *The number of phrases $c(n)$ in a distinct parsing of a binary sequence $X_1, X_2, \ldots, X_n$ satisfies*

$$c(n) \leq \frac{n}{(1-\epsilon_n)\log n}, \tag{13.84}$$

where $\epsilon_n = \min\{1, \frac{\log(\log n)+4}{\log n}\} \to 0$ *as* $n \to \infty$.

Proof: Let

$$n_k = \sum_{j=1}^{k} j2^j = (k-1)2^{k+1} + 2 \tag{13.85}$$

be the sum of the lengths of all distinct strings of length less than or equal to k. The number of phrases c in a distinct parsing of a sequence of length n is maximized when all the phrases are as short as possible. If $n = n_k$, this occurs when all the phrases are of length $\leq k$, and thus

$$c(n_k) \leq \sum_{j=1}^{k} 2^j = 2^{k+1} - 2 < 2^{k+1} \leq \frac{n_k}{k-1}. \tag{13.86}$$

If $n_k \leq n < n_{k+1}$, we write $n = n_k + \Delta$, where $\Delta < (k+1)2^{k+1}$. Then the parsing into shortest phrases has each of the phrases of length $\leq k$ and $\Delta/(k+1)$ phrases of length $k+1$. Thus,

$$c(n) \leq \frac{n_k}{k-1} + \frac{\Delta}{k+1} \leq \frac{n_k + \Delta}{k-1} = \frac{n}{k-1}. \tag{13.87}$$

We now bound the size of k for a given n. Let $n_k \leq n < n_{k+1}$. Then

$$n \geq n_k = (k-1)2^{k+1} + 2 \geq 2^k, \tag{13.88}$$

and therefore

$$k \leq \log n. \tag{13.89}$$

Moreover,

$$n \leq n_{k+1} = k2^{k+2} + 2 \leq (k+2)2^{k+2} \leq (\log n + 2)2^{k+2}, \tag{13.90}$$

by (13.89), and therefore

$$k + 2 \geq \log \frac{n}{\log n + 2}, \tag{13.91}$$

or for all $n \geq 4$,

$$k - 1 \geq \log n - \log(\log n + 2) - 3 \tag{13.92}$$

$$= \left(1 - \frac{\log(\log n + 2) + 3}{\log n}\right) \log n \tag{13.93}$$

$$\geq \left(1 - \frac{\log(2 \log n) + 3}{\log n}\right) \log n \tag{13.94}$$

$$= \left(1 - \frac{\log(\log n) + 4}{\log n}\right) \log n \tag{13.95}$$

$$= (1 - \epsilon_n) \log n. \tag{13.96}$$

Note that $\epsilon_n = \min\{1, \frac{\log(\log n)+4}{\log n}\}$. Combining (13.96) with (13.87), we obtain the lemma. □

We will need a simple result on maximum entropy in the proof of the main theorem.

Lemma 13.5.4 *Let Z be a nonnegative integer-valued random variable with mean μ. Then the entropy $H(Z)$ is bounded by*

$$H(Z) \leq (\mu + 1) \log(\mu + 1) - \mu \log \mu. \tag{13.97}$$

Proof: The lemma follows directly from the results of Theorem 12.1.1, which show that the geometric distribution maximizes the entropy of a nonnegative integer-valued random variable subject to a mean constraint. □

Let $\{X_i\}_{i=-\infty}^{\infty}$ be a binary stationary ergodic process with probability mass function $P(x_1, x_2, \ldots, x_n)$. (Ergodic processes are discussed in greater detail in Section 16.8.) For a fixed integer k, define the kth-order Markov approximation to P as

$$Q_k(x_{-(k-1)}, \ldots, x_0, x_1, \ldots, x_n) \stackrel{\triangle}{=} P(x_{-(k-1)}^0) \prod_{j=1}^{n} P(x_j | x_{j-k}^{j-1}), \tag{13.98}$$

where $x_i^j \stackrel{\triangle}{=} (x_i, x_{i+1}, \ldots, x_j)$, $i \leq j$, and the initial state $x_{-(k-1)}^0$ will be part of the specification of Q_k. Since $P(X_n | X_{n-k}^{n-1})$ is itself an ergodic

process, we have

$$-\frac{1}{n}\log Q_k(X_1, X_2, \ldots, X_n|X_{-(k-1)}^0) = -\frac{1}{n}\sum_{j=1}^{n}\log P(X_j|X_{j-k}^{j-1}) \tag{13.99}$$

$$\rightarrow -E\log P(X_j|X_{j-k}^{j-1}) \tag{13.100}$$

$$= H(X_j|X_{j-k}^{j-1}). \tag{13.101}$$

We will bound the rate of the LZ78 code by the entropy rate of the kth-order Markov approximation for all k. The entropy rate of the Markov approximation $H(X_j|X_{j-k}^{j-1})$ converges to the entropy rate of the process as $k \rightarrow \infty$, and this will prove the result.

Suppose that $X_{-(k-1)}^n = x_{-(k-1)}^n$, and suppose that x_1^n is parsed into c distinct phrases, $y_1, y_2, \ldots, y_c$. Let v_i be the index of the start of the ith phrase (i.e., $y_i = x_{v_i}^{v_{i+1}-1}$). For each $i = 1, 2, \ldots, c$, define $s_i = x_{v_i-k}^{v_i-1}$. Thus, s_i is the k bits of x preceding y_i. Of course, $s_1 = x_{-(k-1)}^0$.

Let c_{ls} be the number of phrases y_i with length l and preceding state $s_i = s$ for $l = 1, 2, \ldots$ and $s \in \mathcal{X}^k$. We then have

$$\sum_{l,s} c_{ls} = c \tag{13.102}$$

and

$$\sum_{l,s} l c_{ls} = n. \tag{13.103}$$

We now prove a surprising upper bound on the probability of a string based on the parsing of the string.

Lemma 13.5.5 (*Ziv's inequality*) *For any distinct parsing (in particular, the LZ78 parsing) of the string $x_1x_2\cdots x_n$, we have*

$$\log Q_k(x_1, x_2, \ldots, x_n|s_1) \leq -\sum_{l,s} c_{ls}\log c_{ls}. \tag{13.104}$$

Note that the right-hand side does not depend on Q_k.

Proof: We write

$$Q_k(x_1, x_2, \ldots, x_n|s_1) = Q_k(y_1, y_2, \ldots, y_c|s_1) \tag{13.105}$$

$$= \prod_{i=1}^{c} P(y_i|s_i) \tag{13.106}$$

or

$$\log Q_k(x_1, x_2, \ldots, x_n|s_1) = \sum_{i=1}^{c} \log P(y_i|s_i) \tag{13.107}$$

$$= \sum_{l,s} \sum_{i:|y_i|=l, s_i=s} \log P(y_i|s_i) \tag{13.108}$$

$$= \sum_{l,s} c_{ls} \sum_{i:|y_i|=l, s_i=s} \frac{1}{c_{ls}} \log P(y_i|s_i) \tag{13.109}$$

$$\leq \sum_{l,s} c_{ls} \log \left(\sum_{i:|y_i|=l, s_i=s} \frac{1}{c_{ls}} P(y_i|s_i) \right), \tag{13.110}$$

where the inequality follows from Jensen's inequality and the concavity of the logarithm.

Now since the y_i are distinct, we have $\sum_{i:|y_i|=l, s_i=s} P(y_i|s_i) \leq 1$. Thus,

$$\log Q_k(x_1, x_2, \ldots, x_n|s_1) \leq \sum_{l,s} c_{ls} \log \frac{1}{c_{ls}}, \tag{13.111}$$

proving the lemma. □

We can now prove the main theorem.

Theorem 13.5.2 *Let $\{X_n\}$ be a binary stationary ergodic process with entropy rate $H(\mathcal{X})$, and let $c(n)$ be the number of phrases in a distinct parsing of a sample of length n from this process. Then*

$$\limsup_{n\to\infty} \frac{c(n) \log c(n)}{n} \leq H(\mathcal{X}) \tag{13.112}$$

with probability 1.

Proof: We begin with Ziv's inequality, which we rewrite as

$$\log Q_k(x_1, x_2, \ldots, x_n|s_1) \leq -\sum_{l,s} c_{ls} \log \frac{c_{ls} c}{c} \tag{13.113}$$

$$= -c \log c - c \sum_{ls} \frac{c_{ls}}{c} \log \frac{c_{ls}}{c}. \tag{13.114}$$

Writing $\pi_{ls} = \frac{c_{ls}}{c}$, we have

$$\sum_{l,s} \pi_{ls} = 1, \qquad \sum_{l,s} l\pi_{ls} = \frac{n}{c}, \tag{13.115}$$

from (13.102) and (13.103). We now define random variables U, V such that

$$\Pr(U = l, V = s) = \pi_{ls}. \tag{13.116}$$

Thus, $EU = \frac{n}{c}$ and

$$\log Q_k(x_1, x_2, \ldots, x_n | s_1) \le cH(U, V) - c \log c \tag{13.117}$$

or

$$-\frac{1}{n} \log Q_k(x_1, x_2, \ldots, x_n | s_1) \ge \frac{c}{n} \log c - \frac{c}{n} H(U, V). \tag{13.118}$$

Now

$$H(U, V) \le H(U) + H(V) \tag{13.119}$$

and $H(V) \le \log |\mathcal{X}|^k = k$. By Lemma 13.5.4, we have

$$H(U) \le (EU + 1) \log(EU + 1) - (EU) \log(EU) \tag{13.120}$$

$$= \left(\frac{n}{c} + 1\right) \log \left(\frac{n}{c} + 1\right) - \frac{n}{c} \log \frac{n}{c} \tag{13.121}$$

$$= \log \frac{n}{c} + \left(\frac{n}{c} + 1\right) \log \left(\frac{c}{n} + 1\right). \tag{13.122}$$

Thus,

$$\frac{c}{n} H(U, V) \le \frac{c}{n} k + \frac{c}{n} \log \frac{n}{c} + o(1). \tag{13.123}$$

For a given n, the maximum of $\frac{c}{n} \log \frac{n}{c}$ is attained for the maximum value of c (for $\frac{c}{n} \le \frac{1}{e}$). But from Lemma 13.5.3, $c \le \frac{n}{\log n}(1 + o(1))$. Thus,

$$\frac{c}{n} \log \frac{n}{c} \le O\left(\frac{\log \log n}{\log n}\right), \tag{13.124}$$

and therefore $\frac{c}{n}H(U, V) \to 0$ as $n \to \infty$. Therefore,

$$\frac{c(n)\log c(n)}{n} \le -\frac{1}{n}\log Q_k(x_1, x_2, \ldots, x_n|s_1) + \epsilon_k(n), \tag{13.125}$$

where $\epsilon_k(n) \to 0$ as $n \to \infty$. Hence, with probability 1,

$$\limsup_{n\to\infty} \frac{c(n)\log c(n)}{n} \le \lim_{n\to\infty} -\frac{1}{n}\log Q_k(X_1, X_2, \ldots, X_n|X_{-(k-1)}^0) \tag{13.126}$$

$$= H(X_0|X_{-1}, \ldots, X_{-k}) \tag{13.127}$$

$$\to H(\mathcal{X}) \qquad \text{as } k \to \infty. \quad \square \tag{13.128}$$

We now prove that LZ78 coding is asymptotically optimal.

Theorem 13.5.3 *Let $\{X_i\}_{-\infty}^{\infty}$ be a binary stationary ergodic stochastic process. Let $l(X_1, X_2, \ldots, X_n)$ be the LZ78 codeword length associated with $X_1, X_2, \ldots, X_n$. Then*

$$\limsup_{n\to\infty} \frac{1}{n} l(X_1, X_2, \ldots, X_n) \le H(\mathcal{X}) \quad \textit{with probability 1}, \tag{13.129}$$

where $H(\mathcal{X})$ is the entropy rate of the process.

Proof: We have shown that $l(X_1, X_2, \ldots, X_n) = c(n)(\log c(n) + 1)$, where $c(n)$ is the number of phrases in the LZ78 parsing of the string $X_1, X_2, \ldots, X_n$. By Lemma 13.5.3, $\limsup c(n)/n = 0$, and thus Theorem 13.5.2 establishes that

$$\limsup \frac{l(X_1, X_2, \ldots, X_n)}{n} = \limsup \left(\frac{c(n)\log c(n)}{n} + \frac{c(n)}{n} \right)$$

$$\le H(\mathcal{X}) \quad \text{with probability 1.} \ \square \tag{13.130}$$

Thus, the length per source symbol of the LZ78 encoding of an ergodic source is asymptotically no greater than the entropy rate of the source. There are some interesting features of the proof of the optimality of LZ78 that are worth noting. The bounds on the number of distinct phrases and Ziv's inequality apply to any distinct parsing of the string, not just the incremental parsing version used in the algorithm. The proof can be extended in many ways with variations on the parsing algorithm; for example, it is possible to use multiple trees that are context or state

dependent [218, 426]. Ziv's inequality (Lemma 13.5.5) remains particularly intriguing since it relates a probability on one side with a purely deterministic function of the parsing of a sequence on the other.

The Lempel–Ziv codes are simple examples of a universal code (i.e., a code that does not depend on the distribution of the source). This code can be used without knowledge of the source distribution and yet will achieve an asymptotic compression equal to the entropy rate of the source.

SUMMARY

Ideal word length

$$l^*(x) = \log \frac{1}{p(x)}. \tag{13.131}$$

Average description length

$$E_p l^*(x) = H(p). \tag{13.132}$$

Estimated probability distribution $\hat{p}(x)$. If $\hat{l}(x) = \log \frac{1}{\hat{p}(x)}$, then

$$E_p \hat{l}(x) = H(p) + D(p||\hat{p}). \tag{13.133}$$

Average redundancy

$$R_p = E_p l(X) - H(p). \tag{13.134}$$

Minimax redundancy. For $X \sim p_\theta(x)$, $\theta \in \theta$,

$$D^* = \min_l \max_p R_p = \min_q \max_\theta D(p_\theta||q). \tag{13.135}$$

Minimax theorem. $D^* = C$, where C is the capacity of the channel $\{\theta, p_\theta(x), \mathcal{X}\}$.

Bernoulli sequences. For $X^n \sim$Bernoulli(θ), the redundancy is

$$D_n^* = \min_q \max_\theta D(p_\theta(x^n)||q(x^n)) \approx \frac{1}{2}\log n + o(\log n). \tag{13.136}$$

Arithmetic coding. nH bits of $F(x^n)$ reveal approximately n bits of x^n.

Lempel–Ziv coding (recurrence time coding). Let $R_n(X^n)$ be the last time in the past that we have seen a block of n symbols X^n. Then $\frac{1}{n}\log R_n \to H(\mathcal{X})$, and encoding by describing the recurrence time is asymptotically optimal.

Lempel–Ziv coding (sequence parsing). If a sequence is parsed into the shortest phrases not seen before (e.g., 011011101 is parsed to 0,1,10,11,101,...) and $l(x^n)$ is the description length of the parsed sequence, then

$$\limsup \frac{1}{n} l(X^n) \le H(\mathcal{X}) \qquad \text{with probability 1} \tag{13.137}$$

for every stationary ergodic process $\{X_i\}$.

PROBLEMS

13.1 *Minimax regret data compression and channel capacity*. First consider universal data compression with respect to two source distributions. Let the alphabet $V = \{1, e, 0\}$ and let $p_1(v)$ put mass $1-\alpha$ on $v = 1$ and mass α on $v = e$. Let $p_2(v)$ put mass $1-\alpha$ on 0 and mass α on $v = e$. We assign word lengths to V according to $l(v) = \log \frac{1}{p(v)}$, the ideal codeword length with respect to a cleverly chosen probability mass function $p(v)$. The worst-case excess description length (above the entropy of the true distribution) is

$$\max_i \left(E_{p_i} \log \frac{1}{p(V)} - E_{p_i} \log \frac{1}{p_i(V)} \right) = \max_i D(p_i \parallel p). \tag{13.138}$$

Thus, the minimax regret is $D^* = \min_p \max_i D(p_i \parallel p)$.

(a) Find D^*.

(b) Find the $p(v)$ achieving D^*.

(c) Compare D^* to the capacity of the binary erasure channel

$$\begin{bmatrix} 1-\alpha & \alpha & 0 \\ 0 & \alpha & 1-\alpha \end{bmatrix}$$

and comment.

13.2 *Universal data compression*. Consider three possible source distributions on $\mathcal{X}$,

$$P_a = (0.7, 0.2, 0.1), \quad P_b = (0.1, 0.7, 0.2), \quad P_c = (0.2, 0.1, 0.7).$$

(a) Find the minimum incremental cost of compression

$$D^* = \min_P \max_\theta D(P_\theta \| P),$$

the associated mass function $P = (p_1, p_2, p_3)$, and ideal codeword lengths $l_i = \log(1/p_i)$.

(b) What is the channel capacity of a channel matrix with rows P_a, P_b, P_c?

13.3 *Arithmetic coding*. Let $\{X_i\}_{i=0}^{\infty}$ be a stationary binary Markov chain with transition matrix

$$p_{ij} = \begin{bmatrix} \frac{3}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix}. \tag{13.139}$$

Calculate the first 3 bits of $F(X^\infty) = 0.F_1F_2\ldots$ when $X^\infty = 1010111\ldots$. How many bits of X^∞ does this specify?

13.4 *Arithmetic coding*. Let X_i be binary stationary Markov with transition matrix $\begin{bmatrix} \frac{1}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix}$.

(a) Find $F(01110) = \Pr\{.X_1X_2X_3X_4X_5 < .01110\}$.

(b) How many bits $.F_1F_2\ldots$ can be known for sure if it is not known how $X = 01110$ continues?

13.5 *Lempel–Ziv*. Give the LZ78 parsing and encoding of 000000110101000000110101.

13.6 *Compression of constant sequence*. We are given the constant sequence $x^n = 11111\ldots$.

(a) Give the LZ78 parsing for this sequence.

(b) Argue that the number of encoding bits per symbol for this sequence goes to zero as $n \to \infty$.

13.7 *Another idealized version of Lempel–Ziv coding*. An idealized version of LZ was shown to be optimal: The encoder and decoder both have available to them the "infinite past" generated by the process, $\ldots, X_{-1}, X_0$, and the encoder describes the string $(X_1, X_2, \ldots, X_n)$ by telling the decoder the position R_n in the past

of the first recurrence of that string. This takes roughly $\log R_n + 2 \log \log R_n$ bits. Now consider the following variant: Instead of describing R_n, the encoder describes R_{n-1} plus the last symbol, X_n. From these two the decoder can reconstruct the string $(X_1, X_2, \ldots, X_n)$.

(a) What is the number of bits per symbol used in this case to encode $(X_1, X_2, \ldots, X_n)$?

(b) Modify the proof given in the text to show that this version is also asymptotically optimal: namely, that the expected number of bits per symbol converges to the entropy rate.

13.8 *Length of pointers in LZ77*. In the version of LZ77 due to Storer and Szymanski [507] described in Section 13.4.1, a short match can be represented by either (F, P, L) (flag, pointer, length) or by (F, C) (flag, character). Assume that the window length is W, and assume that the maximum match length is M.

(a) How many bits are required to represent P? To represent L?

(b) Assume that C, the representation of a character, is 8 bits long. If the representation of P plus L is longer than 8 bits, it would be better to represent a single character match as an uncompressed character rather than as a match within the dictionary. As a function of W and M, what is the shortest match that one should represent as a match rather than as uncompressed characters?

(c) Let $W = 4096$ and $M = 256$. What is the shortest match that one would represent as a match rather than uncompressed characters?

13.9 *Lempel–Ziv 78*.

(a) Continue the Lempel–Ziv parsing of the sequence 0,00,001,00000011010111.

(b) Give a sequence for which the number of phrases in the LZ parsing grows as fast as possible.

(c) Give a sequence for which the number of phrases in the LZ parsing grows as slowly as possible.

13.10 *Two versions of fixed-database Lempel–Ziv*. Consider a source $(\mathcal{A}, P)$. For simplicity assume that the alphabet is finite $|\mathcal{A}| = A < \infty$ and the symbols are i.i.d. $\sim P$. A fixed database $\mathcal{D}$ is given and is revealed to the decoder. The encoder parses the target sequence x_1^n into blocks of length l, and subsequently encodes them by giving the binary description of their last appearance

in the database. If a match is not found, the entire block is sent uncompressed, requiring $l \log A$ bits. A flag is used to tell the decoder whether a match location is being described or the sequence itself. Parts (a) and (b) give some preliminaries you will need in showing the optimality of fixed-database LZ in part (c).

(a) Let x_l be a δ-typical sequence of length l starting at 0, and let $R_l(x^l)$ be the corresponding recurrence index in the infinite past $\ldots, X_{-2}, X_{-1}$. Show that

$$E\big[R_l(X^l)|X^l = x^l\big] \leq 2^{l(H+\delta)},$$

where H is the entropy rate of the source.

(b) Prove that for any $\epsilon > 0$, $\Pr\big(R_l(X^l) > 2^{l(H+\epsilon)}\big) \to 0$ as $l \to \infty$. (*Hint:* Expand the probability by conditioning on strings x^l, and break things up into typical and nontypical. Markov's inequality and the AEP should prove handy as well.)

(c) Consider the following two fixed databases: (i) $\mathcal{D}_1$ is formed by taking all δ-typical l-vectors; and (ii) $\mathcal{D}_2$ formed by taking the most recent $\tilde{L} = 2^{l(H+\delta)}$ symbols in the infinite past (i.e., $X_{-\tilde{L}}, \ldots, X_{-1}$). Argue that the algorithm described above is asymptotically optimal: namely, that the expected number of bits per symbol converges to the entropy rate when used in conjunction with either database $\mathcal{D}_1$ or $\mathcal{D}_2$.

13.11 *Tunstall coding*. The normal setting for source coding maps a symbol (or a block of symbols) from a finite alphabet onto a variable-length string. An example of such a code is the Huffman code, which is the optimal (minimal expected length) mapping from a set of symbols to a prefix-free set of codewords. Now consider the dual problem of variable-to-fixed length codes, where we map a variable-length sequence of source symbols into a fixed-length binary (or D-ary) representation. A variable-to-fixed length code for an i.i.d. sequence of random variables $X_1, X_2, \ldots, X_n$, $X_i \sim p(x)$, $x \in \mathcal{X} = \{0, 1, \ldots, m-1\}$, is defined by a prefix-free set of phrases $A_D \subset \mathcal{X}^*$, where $\mathcal{X}^*$ is the set of finite-length strings of symbols of $\mathcal{X}$, and $|A_D| = D$. Given any sequence $X_1, X_2, \ldots, X_n$, the string is parsed into phrases from A_D (unique because of the prefix-free property of A_D) and represented by a sequence of symbols from a D-ary alphabet. Define the efficiency of this coding scheme by

$$R(A_D) = \frac{\log D}{EL(A_D)}, \tag{13.140}$$

where $EL(A_D)$ is the expected length of a phrase from A_D.

(a) Prove that $R(A_D) \geq H(X)$.

(b) The process of constructing A_D can be considered as a process of constructing an m-ary tree whose leaves are the phrases in A_D. Assume that $D = 1 + k(m-1)$ for some integer $k \geq 1$. Consider the following algorithm due to Tunstall:

(i) Start with $A = \{0, 1, \ldots, m-1\}$ with probabilities p_0, $p_1, \ldots, p_{m-1}$. This corresponds to a complete m-ary tree of depth 1.

(ii) Expand the node with the highest probability. For example, if p_0 is the node with the highest probability, the new set is $A = \{00, 01, \ldots, 0(m-1), 1, \ldots, (m-1)\}$.

(iii) Repeat step 2 until the number of leaves (number of phrases) reaches the required value.

Show that the Tunstall algorithm is optimal in the sense that it constructs a variable to a fixed code with the best $R(A_D)$ for a given D [i.e., the largest value of $EL(A_D)$ for a given D].

(c) Show that there exists a D such that $R(A_D^*) < H(X) + 1$.

HISTORICAL NOTES

The problem of encoding a source with an unknown distribution was analyzed by Fitingof [211] and Davisson [159], who showed that there were classes of sources for which the universal coding procedure was asymptotically optimal. The result relating the average redundancy of a universal code and channel capacity is due to Gallager [229] and Ryabko [450]. Our proof follows that of Csiszár. This result was extended to show that the channel capacity was the lower bound for the redundancy for "most" sources in the class by Merhav and Feder [387], extending the results obtained by Rissanen [444, 448] for the parametric case.

The arithmetic coding procedure has its roots in the Shannon–Fano code developed by Elias (unpublished), which was analyzed by Jelinek [297]. The procedure for the construction of a prefix-free code described in the text is due to Gilbert and Moore [249]. Arithmetic coding itself was developed by Rissanen [441] and Pasco [414]; it was generalized by Langdon and Rissanen [343]. See also the enumerative methods in Cover [120]. Tutorial introductions to arithmetic coding can be found in Langdon [342] and Witten et al. [564]. Arithmetic coding combined with the context-tree weighting algorithm due to Willems et al. [560, 561] achieve the Rissanen

lower bound [444] and therefore have the optimal rate of convergence to the entropy for tree sources with unknown parameters.

The class of Lempel–Ziv algorithms was first described in the seminal papers of Lempel and Ziv [603, 604]. The original results were theoretically interesting, but people implementing compression algorithms did not take notice until the publication of a simple efficient version of the algorithm due to Welch [554]. Since then, multiple versions of the algorithms have been described, many of them patented. Versions of this algorithm are now used in many compression products, including GIF files for image compression and the CCITT standard for compression in modems. The optimality of the sliding window version of Lempel–Ziv (LZ77) is due to Wyner and Ziv [575]. An extension of the proof of the optimality of LZ78 [426] shows that the redundancy of LZ78 is on the order of $1/\log(n)$, as opposed to the lower bounds of $\log(n)/n$. Thus even though LZ78 is asymptotically optimal for all stationary ergodic sources, it converges to the entropy rate very slowly compared to the lower bounds for finite-state Markov sources. However, for the class of all ergodic sources, lower bounds on the redundancy of a universal code do not exist, as shown by examples due to Shields [492] and Shields and Weiss [494]. A lossless block compression algorithm based on sorting the blocks and using simple run-length encoding due to Burrows and Wheeler [81] has been analyzed by Effros et al. [181]. Universal methods for prediction are discussed in Feder, Merhav and Gutman [204, 386, 388].

CHAPTER 14

KOLMOGOROV COMPLEXITY

The great mathematician Kolmogorov culminated a lifetime of research in mathematics, complexity, and information theory with his definition in 1965 of the intrinsic descriptive complexity of an object. In our treatment so far, the object X has been a random variable drawn according to a probability mass function $p(x)$. If X is random, there is a sense in which the descriptive complexity of the event $X = x$ is $\log \frac{1}{p(x)}$, because $\lceil \log \frac{1}{p(x)} \rceil$ is the number of bits required to describe x by a Shannon code. One notes immediately that the descriptive complexity of such an object depends on the probability distribution.

Kolmogorov went further. He defined the algorithmic (descriptive) complexity of an object to be the length of the shortest binary computer program that describes the object. (Apparently, a computer, the most general form of data decompressor, will after a finite amount of computation, use this description to exhibit the object described.) Thus, the Kolmogorov complexity of an object dispenses with the probability distribution. Kolmogorov made the crucial observation that the definition of complexity is essentially computer independent. It is an amazing fact that the expected length of the shortest binary computer description of a random variable is approximately equal to its entropy. Thus, the shortest computer description acts as a universal code which is uniformly good for all probability distributions. In this sense, algorithmic complexity is a conceptual precursor to entropy.

Perhaps a good point of view of the role of this chapter is to consider Kolmogorov complexity as a way to think. One does not use the shortest computer program in practice because it may take infinitely long to find such a minimal program. But one can use very short, not necessarily minimal programs in practice; and the idea of finding such short programs leads to universal codes, a good basis for inductive inference, a formalization of Occam's razor ("The simplest explanation is best") and to fundamental understanding in physics, computer science, and communication theory.

Elements of Information Theory, Second Edition, By Thomas M. Cover and Joy A. Thomas

Before formalizing the notion of Kolmogorov complexity, let us give three strings as examples:

1. 01010101010101010101010101010101010101010101010101010101010101010101
2. 01101010000010011110011001100111111100111011110011001001000010000
3. 11011110011101011111011011111011101011011110000101110010100111011

What are the shortest binary computer programs for each of these sequences? The first sequence is definitely simple. It consists of thirty-two 01's. The second sequence looks random and passes most tests for randomness, but it is in fact the initial segment of the binary expansion of $\sqrt{2}-1$. Again, this is a simple sequence. The third again looks random, except that the proportion of 1's is not near $\frac{1}{2}$. We shall assume that it is otherwise random. It turns out that by describing the number k of 1's in the sequence, then giving the index of the sequence in a lexicographic ordering of those with this number of 1's, one can give a description of the sequence in roughly $\log n + nH(\frac{k}{n})$ bits. This again is substantially fewer than the n bits in the sequence. Again, we conclude that the sequence, random though it is, is simple. In this case, however, it is not as simple as the other two sequences, which have constant-length programs. In fact, its complexity is proportional to n. Finally, we can imagine a truly random sequence generated by pure coin flips. There are 2^n such sequences and they are all equally probable. It is highly likely that such a random sequence cannot be compressed (i.e., there is no better program for such a sequence than simply saying "Print the following: 0101100111010...0"). The reason for this is that there are not enough short programs to go around. Thus, the descriptive complexity of a truly random binary sequence is as long as the sequence itself.

These are the basic ideas. It will remain to be shown that this notion of intrinsic complexity is computer independent (i.e., that the length of the shortest program does not depend on the computer). At first, this seems like nonsense. But it turns out to be true, up to an additive constant. And for long sequences of high complexity, this additive constant (which is the length of the preprogram that allows one computer to mimic the other) is negligible.

14.1 MODELS OF COMPUTATION

To formalize the notions of algorithmic complexity, we first discuss acceptable models for computers. All but the most trivial computers are universal, in the sense that they can mimic the actions of other computers.

We touch briefly on a certain canonical universal computer, the *universal Turing machine*, the conceptually simplest universal computer.

In 1936, Turing was obsessed with the question of whether the thoughts in a living brain could be held equally well by a collection of inanimate parts. In short, could a machine think? By analyzing the human computational process, he posited some constraints on such a computer. Apparently, a human thinks, writes, thinks some more, writes, and so on. Consider a computer as a finite-state machine operating on a finite symbol set. (The symbols in an infinite symbol set cannot be distinguished in finite space.) A program tape, on which a binary program is written, is fed left to right into this finite-state machine. At each unit of time, the machine inspects the program tape, writes some symbols on a work tape, changes its state according to its transition table, and calls for more program. The operations of such a machine can be described by a finite list of transitions. Turing argued that this machine could mimic the computational ability of a human being.

After Turing's work, it turned out that every new computational system could be reduced to a Turing machine, and conversely. In particular, the familiar digital computer with its CPU, memory, and input output devices could be simulated by and could simulate a Turing machine. This led Church to state what is now known as *Church's thesis*, which states that all (sufficiently complex) computational models are equivalent in the sense that they can compute the same family of functions. The class of functions they can compute agrees with our intuitive notion of effectively computable functions, that is, functions for which there is a finite prescription or program that will lead in a finite number of mechanically specified computational steps to the desired computational result.

We shall have in mind throughout this chapter the computer illustrated in Figure 14.1. At each step of the computation, the computer reads a symbol from the input tape, changes state according to its state transition table, possibly writes something on the work tape or output tape, and

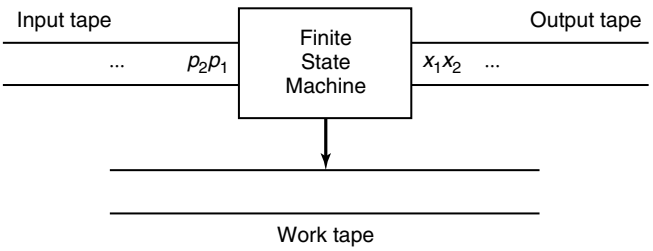

FIGURE 14.1. A Turing machine.

moves the program read head to the next cell of the program read tape. This machine reads the program from right to left only, never going back, and therefore the programs form a prefix-free set. No program leading to a halting computation can be the prefix of another such program. The restriction to prefix-free programs leads immediately to a theory of Kolmogorov complexity which is formally analogous to information theory.

We can view the Turing machine as a map from a set of finite-length binary strings to the set of finite- or infinite-length binary strings. In some cases, the computation does not halt, and in such cases the value of the function is said to be *undefined*. The set of functions $f : \{0, 1\}^* \rightarrow \{0, 1\}^* \cup \{0, 1\}^\infty$ computable by Turing machines is called the set of *partial recursive functions*.

14.2 KOLMOGOROV COMPLEXITY: DEFINITIONS AND EXAMPLES

Let x be a finite-length binary string and let $\mathcal{U}$ be a universal computer. Let $l(x)$ denote the length of the string x. Let $\mathcal{U}(p)$ denote the output of the computer $\mathcal{U}$ when presented with a program p.

We define the Kolmogorov (or algorithmic) complexity of a string x as the minimal description length of x.

Definition The *Kolmogorov complexity* $K_{\mathcal{U}}(x)$ of a string x with respect to a universal computer $\mathcal{U}$ is defined as

$$K_{\mathcal{U}}(x) = \min_{p:\mathcal{U}(p)=x} l(p), \tag{14.1}$$

the minimum length over all programs that print x and halt. Thus, $K_{\mathcal{U}}(x)$ is the shortest description length of x over all descriptions interpreted by computer $\mathcal{U}$.

A useful technique for thinking about Kolmogorov complexity is the following—if one person can describe a sequence to another person in such a manner as to lead unambiguously to a computation of that sequence in a finite amount of time, the number of bits in that communication is an upper bound on the Kolmogorov complexity. For example, one can say "Print out the first 1,239,875,981,825,931 bits of the square root of e." Allowing 8 bits per character (ASCII), we see that the unambiguous 73-symbol program above demonstrates that the Kolmogorov complexity of this huge number is no greater than $(8)(73) = 584$ bits. Most numbers of this length (more than a quadrillion bits) have a Kolmogorov complexity

of nearly 1,239,875,981,825,931 bits. The fact that there is a simple algorithm to calculate the square root of e provides the saving in descriptive complexity.

In the definition above, we have not mentioned anything about the length of x. If we assume that the computer already knows the length of x, we can define the *conditional Kolmogorov complexity* knowing $l(x)$ as

$$K_{\mathcal{U}}(x|l(x)) = \min_{p:\mathcal{U}(p,l(x))=x} l(p). \tag{14.2}$$

This is the shortest description length if the computer $\mathcal{U}$ has the length of x made available to it.

It should be noted that $K_{\mathcal{U}}(x|y)$ is usually defined as $K_{\mathcal{U}}(x|y, y^*)$, where y^* is the shortest program for y. This is to avoid certain slight asymmetries, but we will not use this definition here.

We first prove some of the basic properties of Kolmogorov complexity and then consider various examples.

Theorem 14.2.1 (*Universality of Kolmogorov complexity*) *If $\mathcal{U}$ is a universal computer, for any other computer $\mathcal{A}$ there exists a constant $c_{\mathcal{A}}$ such that*

$$K_{\mathcal{U}}(x) \leq K_{\mathcal{A}}(x) + c_{\mathcal{A}} \tag{14.3}$$

for all strings $x \in \{0, 1\}^*$, and the constant $c_{\mathcal{A}}$ does not depend on x.

Proof: Assume that we have a program $p_{\mathcal{A}}$ for computer $\mathcal{A}$ to print x. Thus, $\mathcal{A}(p_{\mathcal{A}}) = x$. We can precede this program by a simulation program $s_{\mathcal{A}}$ which tells computer $\mathcal{U}$ how to simulate computer $\mathcal{A}$. Computer $\mathcal{U}$ will then interpret the instructions in the program for $\mathcal{A}$, perform the corresponding calculations and print out x. The program for $\mathcal{U}$ is $p = s_{\mathcal{A}} p_{\mathcal{A}}$ and its length is

$$l(p) = l(s_{\mathcal{A}}) + l(p_{\mathcal{A}}) = c_{\mathcal{A}} + l(p_{\mathcal{A}}), \tag{14.4}$$

where $c_{\mathcal{A}}$ is the length of the simulation program. Hence,

$$K_{\mathcal{U}}(x) = \min_{p:\mathcal{U}(p)=x} l(p) \leq \min_{p:\mathcal{A}(p)=x} (l(p) + c_{\mathcal{A}}) = K_{\mathcal{A}}(x) + c_{\mathcal{A}} \tag{14.5}$$

for all strings x. □

The constant $c_{\mathcal{A}}$ in the theorem may be very large. For example, $\mathcal{A}$ may be a large computer with a large number of functions built into the system.

The computer $\mathcal{U}$ can be a simple microprocessor. The simulation program will contain the details of the implementation of all these functions, in fact, all the software available on the large computer. The crucial point is that the length of this simulation program is independent of the length of x, the string to be compressed. For sufficiently long x, the length of this simulation program can be neglected, and we can discuss Kolmogorov complexity without talking about the constants.

If $\mathcal{A}$ and $\mathcal{U}$ are both universal, we have

$$|K_{\mathcal{U}}(x) - K_{\mathcal{A}}(x)| < c \tag{14.6}$$

for all x. Hence, we will drop all mention of $\mathcal{U}$ in all further definitions. We will assume that the unspecified computer $\mathcal{U}$ is a fixed universal computer.

Theorem 14.2.2 (*Conditional complexity is less than the length of the sequence*)

$$K(x|l(x)) \leq l(x) + c. \tag{14.7}$$

Proof: A program for printing x is

Print the following l-bit sequence: $x_1 x_2 \ldots x_{l(x)}$.

Note that no bits are required to describe l since l is given. The program is self-delimiting because $l(x)$ is provided and the end of the program is thus clearly defined. The length of this program is $l(x) + c$. □

Without knowledge of the length of the string, we will need an additional stop symbol or we can use a self-punctuating scheme like the one described in the proof of the next theorem.

Theorem 14.2.3 (*Upper bound on Kolmogorov complexity*)

$$K(x) \leq K(x|l(x)) + 2\log l(x) + c. \tag{14.8}$$

Proof: If the computer does not know $l(x)$, the method of Theorem 14.2.2 does not apply. We must have some way of informing the computer when it has come to the end of the string of bits that describes the sequence. We describe a simple but inefficient method that uses a sequence 01 as a "comma."

Suppose that $l(x) = n$. To describe $l(x)$, repeat every bit of the binary expansion of n twice; then end the description with a 01 so that the computer knows that it has come to the end of the description of n.

For example, the number 5 (binary 101) will be described as 11001101. This description requires $2\lceil \log n \rceil + 2$ bits. Thus, inclusion of the binary representation of $l(x)$ does not add more than $2 \log l(x) + c$ bits to the length of the program, and we have the bound in the theorem. □

A more efficient method for describing n is to do so recursively. We first specify the number $(\log n)$ of bits in the binary representation of n and then specify the actual bits of n. To specify $\log n$, the length of the binary representation of n, we can use the inefficient method $(2 \log \log n)$ or the efficient method $(\log \log n + \cdots)$. If we use the efficient method at each level, until we have a small number to specify, we can describe n in $\log n + \log \log n + \log \log \log n + \cdots$ bits, where we continue the sum until the last positive term. This sum of iterated logarithms is sometimes written $\log^* n$. Thus, Theorem 14.2.3 can be improved to

$$K(x) \leq K(x|l(x)) + \log^* l(x) + c. \tag{14.9}$$

We now prove that there are very few sequences with low complexity.

Theorem 14.2.4 (*Lower bound on Kolmogorov complexity). The number of strings x with complexity $K(x) < k$ satisfies*

$$|\{x \in \{0, 1\}^* : K(x) < k\}| < 2^k. \tag{14.10}$$

Proof: There are not very many short programs. If we list all the programs of length $< k$, we have

$$\underbrace{\Lambda}_{1}, \underbrace{0, 1}_{2}, \underbrace{00, 01, 10, 11}_{4}, \ldots, \underbrace{\ldots, \overbrace{11 \ldots 1}^{k-1}}_{2^{k-1}} \tag{14.11}$$

and the total number of such programs is

$$1 + 2 + 4 + \cdots + 2^{k-1} = 2^k - 1 < 2^k. \tag{14.12}$$

Since each program can produce only one possible output sequence, the number of sequences with complexity $< k$ is less than 2^k. □

To avoid confusion and to facilitate exposition in the rest of this chapter, we shall need to introduce a special notation for the *binary entropy function*

$$H_0(p) = -p \log p - (1 - p) \log(1 - p). \tag{14.13}$$

Thus, when we write $H_0(\frac{1}{n}\sum_{i=1}^{n} X_i)$, we will mean $-\overline{X}_n \log \overline{X}_n - (1 - \overline{X}_n)\log(1 - \overline{X}_n)$ and not the entropy of random variable $\overline{X}_n$. When there is no confusion, we shall simply write $H(p)$ for $H_0(p)$.

Now let us consider various examples of Kolmogorov complexity. The complexity will depend on the computer, but only up to an additive constant. To be specific, we consider a computer that can accept unambiguous commands in English (with numbers given in binary notation). We will use the inequality

$$\sqrt{\frac{n}{8k(n-k)}}2^{nH(k/n)} \leq \binom{n}{k} \leq \sqrt{\frac{n}{\pi k(n-k)}}2^{nH(k/n)}, \quad k \neq 0, n, \tag{14.14}$$

which is proved in Lemma 17.5.1.

Example 14.2.1 (*A sequence of n zeros*) If we assume that the computer knows n, a short program to print this string is

Print the specified number of zeros.

The length of this program is a constant number of bits. This program length does not depend on n. Hence, the Kolmogorov complexity of this sequence is c, and

$$K(000\ldots0|n) = c \qquad \text{for all } n. \tag{14.15}$$

Example 14.2.2 (*Kolmogorov complexity of* π) The first n bits of π can be calculated using a simple series expression. This program has a small constant length if the computer already knows n. Hence,

$$K(\pi_1\pi_2\cdots\pi_n|n) = c. \tag{14.16}$$

Example 14.2.3 (*Gotham weather*) Suppose that we want the computer to print out the weather in Gotham for n days. We can write a program that contains the entire sequence $x = x_1x_2\cdots x_n$, where $x_i = 1$ indicates rain on day i. But this is inefficient, since the weather is quite dependent. We can devise various coding schemes for the sequence to take the dependence into account. A simple one is to find a Markov model to approximate the sequence (using the empirical transition probabilities) and then code the sequence using the Shannon code for this probability distribution. We can describe the empirical Markov transitions in $O(\log n)$ bits and then use $\log \frac{1}{p(x)}$ bits to describe x, where p is the

specified Markov probability. Assuming that the entropy of the weather is $\frac{1}{5}$ bit per day, we can describe the weather for n days using about $n/5$ bits, and hence

$$K(\text{Gotham weather}|n) \approx \frac{n}{5} + O(\log n) + c. \tag{14.17}$$

Example 14.2.4 (*Repeating sequence of the form 01010101...01*) A short program suffices. Simply print the specified number of 01 pairs. Hence,

$$K(010101010\ldots01|n) = c. \tag{14.18}$$

Example 14.2.5 (*Fractal*) A fractal is part of the Mandelbrot set and is generated by a simple computer program. For different points c in the complex plane, one calculates the number of iterations of the map $z_{n+1} = z_n^2 + c$ (starting with $z_0 = 0$) needed for $|z|$ to cross a particular threshold. The point c is then colored according to the number of iterations needed. Thus, the fractal is an example of an object that looks very complex but is essentially very simple. Its Kolmogorov complexity is essentially zero.

Example 14.2.6 (*Mona Lisa*) We can make use of the many structures and dependencies in the painting. We can probably compress the image by a factor of 3 or so by using some existing easily described image compression algorithm. Hence, if n is the number of pixels in the image of the Mona Lisa,

$$K(\text{Mona Lisa}|n) \leq \frac{n}{3} + c. \tag{14.19}$$

Example 14.2.7 (*Integer n*) If the computer knows the number of bits in the binary representation of the integer, we need only provide the values of these bits. This program will have length $c + \log n$.

In general, the computer will not know the length of the binary representation of the integer. So we must inform the computer in some way when the description ends. Using the method to describe integers used to derive (14.9), we see that the Kolmogorov complexity of an integer is bounded by

$$K(n) \leq \log^* n + c. \tag{14.20}$$

Example 14.2.8 (*Sequence of n bits with k ones*) Can we compress a sequence of n bits with k ones?

Our first guess is no, since we have a series of bits that must be reproduced exactly. But consider the following program:

Generate, in lexicographic order, all sequences with k ones;
Of these sequences, print the ith sequence.

This program will print out the required sequence. The only variables in the program are k (with known range $\{0, 1, \ldots, n\}$) and i (with conditional range $\{1, 2, \ldots, \binom{n}{k}\}$). The total length of this program is

$$l(p) = c + \underbrace{\log n}_{\text{to express } k} + \underbrace{\log \binom{n}{k}}_{\text{to express } i} \tag{14.21}$$

$$\leq c' + \log n + nH\left(\tfrac{k}{n}\right) - \tfrac{1}{2}\log n, \tag{14.22}$$

since $\binom{n}{k} \leq \frac{1}{\sqrt{\pi npq}} 2^{nH_0(p)}$ by (14.14) for $p = k/n$ and $q = 1 - p$ and $k \neq 0$ and $k \neq n$. We have used $\log n$ bits to represent k. Thus, if $\sum_{i=1}^{n} x_i = k$, then

$$K(x_1, x_2, \ldots, x_n|n) \leq nH_0\left(\frac{k}{n}\right) + \frac{1}{2}\log n + c. \tag{14.23}$$

We can summarize Example 14.2.8 in the following theorem.

Theorem 14.2.5 *The Kolmogorov complexity of a binary string x is bounded by*

$$K(x_1 x_2 \cdots x_n|n) \leq nH_0\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right) + \frac{1}{2}\log n + c. \tag{14.24}$$

Proof: Use the program described in Example 14.2.8. □

Remark Let $x \in \{0, 1\}^*$ be the data that we wish to compress, and consider the program p to be the compressed data. We will have succeeded in compressing the data only if $l(p) < l(x)$, or

$$K(x) < l(x). \tag{14.25}$$

In general, when the length $l(x)$ of the sequence x is small, the constants that appear in the expressions for the Kolmogorov complexity will overwhelm the contributions due to $l(x)$. Hence, the theory is useful primarily when $l(x)$ is very large. In such cases we can safely neglect the terms that do not depend on $l(x)$.

14.3 KOLMOGOROV COMPLEXITY AND ENTROPY

We now consider the relationship between the Kolmogorov complexity of a sequence of random variables and its entropy. In general, we show that the expected value of the Kolmogorov complexity of a random sequence is close to the Shannon entropy. First, we prove that the program lengths satisfy the Kraft inequality.

Lemma 14.3.1 *For any computer $\mathcal{U}$,*

$$\sum_{p:\mathcal{U}(p)\text{halts}} 2^{-l(p)} \leq 1. \tag{14.26}$$

Proof: If the computer halts on any program, it does not look any further for input. Hence, there cannot be any other halting program with this program as a prefix. Thus, the halting programs form a prefix-free set, and their lengths satisfy the Kraft inequality (Theorem 5.2.1).

We now show that $\frac{1}{n}EK(X^n|n) \approx H(X)$ for i.i.d. processes with a finite alphabet.

Theorem 14.3.1 (*Relationship of Kolmogorov complexity and entropy*) *Let the stochastic process $\{X_i\}$ be drawn i.i.d. according to the probability mass function $f(x)$, $x \in \mathcal{X}$, where $\mathcal{X}$ is a finite alphabet. Let $f(x^n) = \prod_{i=1}^{n} f(x_i)$. Then there exists a constant c such that*

$$H(X) \leq \frac{1}{n}\sum_{x^n} f(x^n)K(x^n|n) \leq H(X) + \frac{(|\mathcal{X}|-1)\log n}{n} + \frac{c}{n} \tag{14.27}$$

for all n. Consequently,

$$E\frac{1}{n}K(X^n|n) \to H(X). \tag{14.28}$$

Proof: Consider the lower bound. The allowed programs satisfy the prefix property, and thus their lengths satisfy the Kraft inequality. We assign to each x^n the length of the shortest program p such that $\mathcal{U}(p, n) = x^n$. These shortest programs also satisfy the Kraft inequality. We know from the theory of source coding that the expected codeword length must be greater than the entropy. Hence,

$$\sum_{x^n} f(x^n)K(x^n|n) \geq H(X_1, X_2, \ldots, X_n) = nH(X). \tag{14.29}$$

We first prove the upper bound when $\mathcal{X}$ is binary (i.e., $X_1, X_2, \ldots, X_n$ are i.i.d. $\sim$ Bernoulli(θ)). Using the method of Theorem 14.2.5, we can bound the complexity of a binary string by

$$K(x_1 x_2 \ldots x_n | n) \leq nH_0\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right) + \frac{1}{2}\log n + c. \tag{14.30}$$

Hence,

$$\begin{aligned} EK(X_1 X_2 \ldots X_n | n) &\leq nEH_0\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) + \frac{1}{2}\log n + c & (14.31)\\ &\overset{(a)}{\leq} nH_0\left(\frac{1}{n}\sum_{i=1}^{n} EX_i\right) + \frac{1}{2}\log n + c & (14.32)\\ &= nH_0(\theta) + \frac{1}{2}\log n + c, & (14.33) \end{aligned}$$

where (a) follows from Jensen's inequality and the concavity of the entropy. Thus, we have proved the upper bound in the theorem for binary processes.

We can use the same technique for the case of a nonbinary finite alphabet. We first describe the type of the sequence (the empirical frequency of occurrence of each of the alphabet symbols as defined in Section 11.1) using $(|\mathcal{X}| - 1)\log n$ bits (the frequency of the last symbol can be calculated from the frequencies of the rest). Then we describe the index of the sequence within the set of all sequences having the same type. The type class has less than $2^{nH(P_{x^n})}$ elements (where P_{x^n} is the type of the sequence x^n) as shown in Chapter 11, and therefore the two-stage description of a string x^n has length

$$K(x^n | n) \leq nH(P_{x^n}) + (|\mathcal{X}| - 1)\log n + c. \tag{14.34}$$

Again, taking the expectation and applying Jensen's inequality as in the binary case, we obtain

$$EK(X^n | n) \leq nH(X) + (|\mathcal{X}| - 1)\log n + c. \tag{14.35}$$

Dividing this by n yields the upper bound of the theorem. □