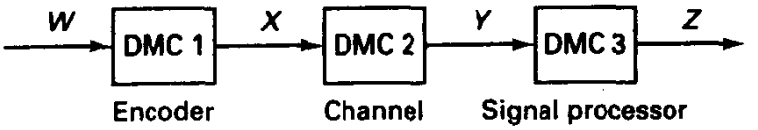


Figure 2.7.7 Cascade of discrete channels.

$I(X;Y)$ or $I(Y;Z)$. To show this, we prove the former: $I(X;Z) - I(X;Y) \leq 0$.

$$\begin{aligned} I(X;Z) - I(X;Y) &= \sum_i \sum_k P(x_i, z_k) \log \left[\frac{P(z_k \mid x_i)}{P(z_k)} \right] \\ &\quad - \sum_i \sum_j P(x_i, y_j) \log \left[\frac{P(y_j \mid x_i)}{P(y_j)} \right] \\ &= \sum_i \sum_j \sum_k P(x_i, y_j, z_k) \log \left[\frac{P(z_k \mid x_i) P(y_j)}{P(z_k) P(y_j \mid x_i)} \right] \\ &\leq \sum_i \sum_j \sum_k P(x_i, y_j, z_k) \frac{\left[\frac{P(z_k \mid x_i) P(y_j)}{P(z_k) P(y_j \mid x_i)} - 1 \right]}{\log_e 2} \end{aligned} \tag{2.7.25}$$

where the last step is by the information theory inequality. Use of Bayes's rule and subsequent summation yield that the final summation is zero. Thus,

$$I(X;Z) - I(X;Y) \leq 0 \tag{2.7.26}$$

A similar development shows $I(X;Z) \leq I(Y;Z)$ in Figure 2.7.7.

We have just demonstrated what is sometimes known as the ***data-processing lemma***, which in essence says that average mutual information cannot be increased by further processing, either deterministic or stochastic. This is somewhat paradoxical, given that communication systems are replete with processors such as quantizers, samplers, and encoders and decoders. The theorem should not imply that these are necessarily harmful, for they often simply manipulate data into another form that preserves information. On the other hand, we must not see such additional processing (or additional channels in cascade) as a way to increase information transfer.

We have developed several results from basic definitions of entropy, channel capacity, and the like. We will now see their special importance to the commun'cation problem. We begin by proving a converse to a coding theorem, to the effect that if message entropy, per channel use, exceeds channel capacity, that *no system* is capable of achieving arbitrarily small error probability. In Section 2.8, we consider the problem of efficiently encoding a discrete memoryless source, where entropy again plays a central role.

2.7.6 Converse to the Noisy Channel Coding Theorem

In Chapter 4, we shall prove the positive side of a coding theorem that guarantees that, if the attempted transmission rate is less than channel capacity, in equal units, then we can, with sufficient effort, make the reliability of transmission arbitrarily good. This proof requires some effort for general channels. It is easy, however, to demonstrate the

converse to the theorem. Our development closely parallels that of Gallager [12], in turn drawn from Fano [14].

Consider the general communication system depicted in Figure 2.7.8, where a discrete memoryless source produces an L-tuple $\mathbf{U} = (U_1, U_2, \ldots, U_L)$ of symbols selected from an alphabet of size K. We allow a completely general encoder to map these strings into codewords for the channel. The channel input vector will be designated as $\mathbf{X} = (X_1, \ldots, X_N)$, while the output vector will be denoted $\mathbf{Y} = (Y_1, \ldots, Y_N)$. The respective alphabet sizes are M and Q as before. The decoder observes $\mathbf{Y}$ and attempts to infer what message sequence was transmitted, and we let $\mathbf{V} = (V_1, \ldots, V_L)$ denote its choice, taken from the same alphabet as that of the source. Maximum likelihood rules would be appropriate for the decoder, as discussed in Section 2.6, but we do not rely on a specific decoding algorithm here.

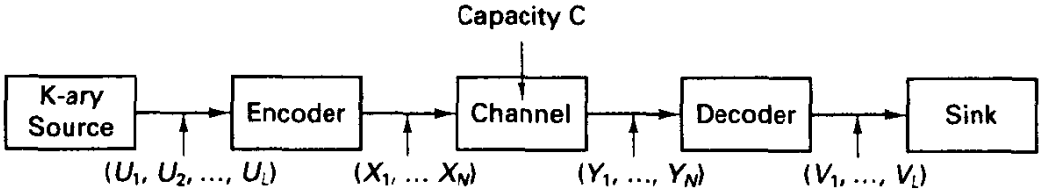


Figure 2.7.8 Block diagram of general information transmission system.

We say a symbol error occurs in the mth position of the message if $U_m \neq V_m$. The probability of error in position m could be expressed as

$$P_{e,m} = \sum_i \sum_j P_{UV}^{(m)}(u_i, v_j)\, \delta(u_i, v_j), \tag{2.7.27}$$

where $\delta(u, v)$ is the Kronecker delta function,

$$\delta(u, v) = \begin{cases} 1, & \text{if } u \neq v, \\ 0, & \text{if } u = v, \end{cases} \tag{2.7.28}$$

and $P_{UV}^{(m)}$ is the joint distribution for the pair (U_m, V_m). The average error probability over L symbols is

$$<P_e> = \frac{1}{L} \sum_{m=1}^{L} P_{e,m}. \tag{2.7.29}$$

Our aim is to show that, for any L, $<P_e>$ is bounded away from zero whenever the entropy of the source, per channel use, exceeds the channel capacity C.

First, we consider the case $L = 1$. We suppose the channel is employed N times to communicate each source symbol. The decoder employs imperfect channel outputs to infer which source symbol was sent, and this variable is denoted V. We first relate the symbol error probability to the conditional entropy $H(U \mid V)$:

Lemma (Fano). The conditional entropy of the source symbol U, given the decoder output symbol V, is bounded by

$$H(U \mid V) \leq P_e \log(K - 1) + h_2(P_e), \tag{2.7.30}$$

where $h_2(x) = -x \log x - (1 - x) \log(1 - x)$ is the binary entropy function introduced in (2.7.5) and P_e is the error probability for a single message symbol.

Proof. The proof relies on a simple application of the grouping axiom for entropy. We consider the uncertainty remaining about U, once V has been observed. We can break the joint ensemble (U, V) into two classes: a *no-error class* in which u and v are identical, and an *error class* in which the input and output do not match. By definition, the latter class, or event, has probability P_e. By the grouping property,

$$\begin{aligned} H(U \mid V) = {} & H(\text{class specification} \mid V) \\ & + P(\text{no error class}) H(U \mid \text{no error class}, V) \\ & + P(\text{error class}) H(U \mid \text{error class}, V). \end{aligned} \tag{2.7.31}$$

The entropy of class specification is that of a binary experiment with parameter P_e. Also, if we are informed that no error occurred, $V = v$ specifies U perfectly, so the second term in (2.7.31) vanishes. On the other hand, if we are informed that an error exists, the remaining uncertainty is, at worst, that of an equiprobable selection among the $K - 1$ symbols not equaling v, namely $\log(K - 1)$. Thus, substitution in (2.7.31) proves the lemma, (2.7.30).

To tie this to source entropy and channel capacity, we note that $H(U \mid V) = H(U) - I(U; V)$ and that $I(U; V) \leq I(\mathbf{X}; \mathbf{Y}) \leq NC$, by the data-processing lemma and by the definition of C. Thus, we have that

$$P_e \log(K - 1) + h_2(P_e) \geq H(U) - NC. \tag{2.7.32}$$

Graphical interpretation of this result is found in Figure 2.7.9, where the solid curve is a plot of the left-hand side of (2.7.32) for some specific K. The right-hand side of (2.7.32) is the difference between entropy and the available capacity, per source symbol. If this is positive, then the smallest achievable P_e is specified by the graphical solution shown. If the right-hand side is negative, the result we obtained does not say perfect communication is possible, only that it *might* be. This positive version of the coding theorem is yet to come.

We now proceed to show that, when source entropy exceeds available capacity per unit source time, encoding $L > 1$ symbols together is no help on a DMC. We return to

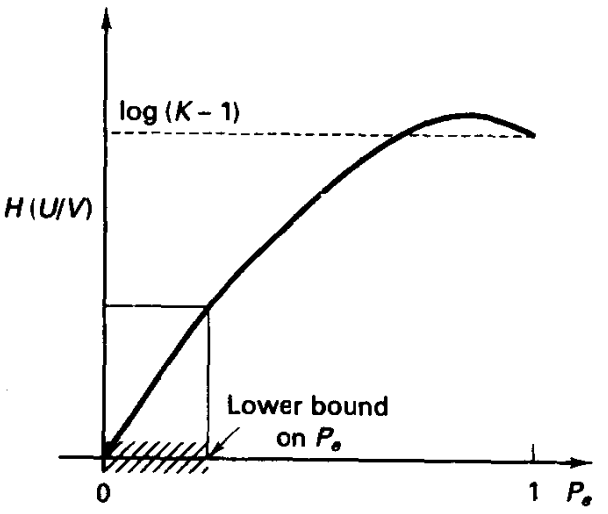


Figure 2.7.9 Interpretation of Fano inequality.

the general L-tuple case and consider the conditional entropy, of $\mathbf{U}$, given $\mathbf{V}$. By the chain rule for entropy,

$$H(\mathbf{U} \mid \mathbf{V}) = H(U_1 \mid \mathbf{V}) + H(U_2 \mid U_1, \mathbf{V}) + \cdots + H(U_L \mid U_1, \ldots, U_{L-1}, \mathbf{V}). \quad (2.7.33)$$

Term-wise application of the conditioning inequality for entropy, $H(X \mid Y) \geq H(X \mid Y, Z)$, in (2.7.33) gives the inequality

$$H(\mathbf{U} \mid \mathbf{V}) \leq \sum_{j=1}^{L} H(U_j \mid V_j). \quad (2.7.34)$$

Now, if we use the Fano lemma (2.7.30) on each term in the sum, we have

$$H(\mathbf{U} \mid \mathbf{V}) \leq \sum_{j=1}^{L} [P_{e,j} \log(K-1) + h_2(P_{e,j})] \quad (2.7.35)$$

or, normalizing by the block length L,

$$\frac{1}{L} H(\mathbf{U} \mid \mathbf{V}) \leq <P_e> \log(K-1) + \frac{1}{L} \sum_{j=1}^{L} h_2(P_{e,j}). \quad (2.7.36)$$

Finally, we invoke the fact that the entropy function is convex $\cap$ over the space of probability assignment vectors [12], illustrated in Figure 2.7.3:

$$\frac{1}{L} \sum_{j=1}^{L} h_2(P_{e,j}) \leq h_2\left(\frac{1}{L} \sum_{j=1}^{L} P_{e,j}\right) = h_2(<P_e>). \quad (2.7.37)$$

Thus, we have

$$\frac{1}{L} H(\mathbf{U} \mid \mathbf{V}) \leq h_2(<P_e>) + <P_e> \log K, \quad (2.7.38)$$

which is similar in form to the result for $L = 1$.

To connect this result to source entropy and channel capacity, we assume that the channel is used N times for every L-tuple from the source. The data-processing lemma proved earlier holds that

$$I(\mathbf{U}; \mathbf{V}) \leq I(\mathbf{X}; \mathbf{Y}) \quad (2.7.39)$$

for any conceivable encoder and decoder. Also,

$$H(\mathbf{U} \mid \mathbf{V}) = H(\mathbf{U}) - I(\mathbf{U}; \mathbf{V}). \quad (2.7.40)$$

Finally, for a memoryless source, $H(\mathbf{U}) = LH(U)$. Substituting in the left-hand side of (2.7.38) yields

$$\begin{aligned} \frac{1}{L} H(\mathbf{U} \mid \mathbf{V}) &= H(U) - \frac{1}{L} I(\mathbf{U}; \mathbf{V}) \\ &\geq H(U) - \frac{1}{L} I(\mathbf{X}; \mathbf{Y}). \end{aligned} \quad (2.7.41)$$

For N uses of a memoryless channel, we have that $I(\mathbf{X}; \mathbf{Y}) \leq NC$, and (2.7.41) together with (2.7.38) yields

$$<P_e> \log(K-1) + h_2(<P_e>) \geq H(U) - \frac{N}{L} C. \quad (2.7.42)$$

The first term on the right in (2.7.42) is the source entropy per symbol, while the second term is the amount of channel capacity available per source symbol. The implication of (2.7.42) is clear with the aid of Figure 2.7.9. If the source entropy exceeds channel capacity, per source symbol, then $<P_e>$ is bounded away from zero. As long as the ratio N/L is kept constant, increasing L does not help in this regard, and there is no possibility of forcing $<P_e>$ to an arbitrarily small number, or $<P_e>$ is strictly bounded away from zero.

Example 2.29 Application of Converse to the Coding Theorem

Suppose we are presented a binary memoryless source whose probabilities are 0.6 and 0.4. Let the source produce 1000 symbols per second. Suppose we have at our disposal a binary erasure channel, with erasure probability $\delta = 0.1$, that can also be utilized 1000 times per second.

The source entropy is

$$\begin{aligned} H(U) &= -0.6\log 0.6 - 0.4\log 0.4 \\ &= 0.971\text{bit/symbol.} \end{aligned} \tag{2.7.43}$$

The channel capacity can be easily computed from (2.7.19b) to be $C = 0.9$ bit per channel use. Since $N/L = 1$ here, the right-hand side of (2.7.42) becomes 0.071. Thus, we seek the solution to

$$<P_e>\log(2-1) + h_2(<P_e>) = 0.071 \tag{2.7.44}$$

which is $<P_e> \geq 0.0085$. The result is that no means of encoding and decoding that keeps $N/L = 1$ can achieve an average error probability $<P_e>$ smaller than 0.0085. (This might be tolerable, of course.)

It is worthwhile considering what a naive transmission system would do. We send 0 or 1 through the BEC dependent on source output, with no coding at all. If an erasure occurs, the decoder flips a fair coin to decide the input. It is obvious then that the symbol error probability is $<P_e> = (0.1)(0.5) = 0.05$. Any more sophisticated scheme cannot attain more than a sixfold improvement in error probability.

As a variation, we might consider the same problem formulation, except allowing the same channel to be used 1200 times per second. (Usually, we cannot increase the transmission speed and maintain the same discrete channel quality without some engineering changes, but let's ignore that issue here.) Now the available capacity per source symbol exceeds the entropy, and (2.7.41) has no positive solution for $<P_e>$. This should not imply that the error probability can be exactly zero, nor should it imply that we can make the error probability arbitrarily small (it turns out we can, but we have yet to show this). We should simply conclude that the converse to the coding theorem does not itself impose a nonzero lower limit on error probability.

2.8 CODING OF DISCRETE INFORMATION SOURCES

Although it will not be a principal topic of the remainder of the book, efficient coding of discrete sources for digital transmission can now be easily appreciated, and the discussion further illustrates the importance of entropy and information measures. This material is variously known as ***discrete source coding***, data compression, data compaction, or noiseless source coding. We emphasize the encoding of memoryless sources and then generalize to Markov sources, an important class of sources with memory.

A ***discrete memoryless source (DMS)*** produces a sequence of output symbols that are selected independently from a K-ary alphabet $\{0, 1, \ldots, K-1\}$ with probabilities $P_0, \ldots, P_{K-1}$. The entropy of the source $H(U)$ is, because of the memoryless property,

$$H(U) = -\sum_{i=0}^{K-1} P_i \log P_i \text{ bits/source symbol.} \tag{2.8.1}$$

Consider a message to be an L-tuple of consecutive outputs from such a source. Such strings have entropy $LH(U)$ bits per L-tuple, again by the memoryless assumption. Our interest is in encoding, or representing, the output string produced by the source in an efficient manner, and we will consider both block and variable-length coding techniques for so doing. The entropy $H(U)$ places a fundamental limit on efficiency in either case. We begin with block source codes.

2.8.1 Block Source Codes

A ***block source code*** is a relation between source L-tuples and codewords of fixed length N from an D-ary alphabet, typically with $D = 2$ so that we have a binary encoding of the source. The D-ary sequences are transmitted (or stored) in lieu of sending the source string directly, and in some sense, we would like to utilize as few code symbols as possible to represent the message. The source decoder is assumed to know the code, in the form of a dictionary or encoding algorithm, and uses the codewords to reproduce the source vector. The entire operation is depicted in Figure 2.8.1. In studies of source coding, it is normally thought that the channel is perfect at transmitting source codewords.

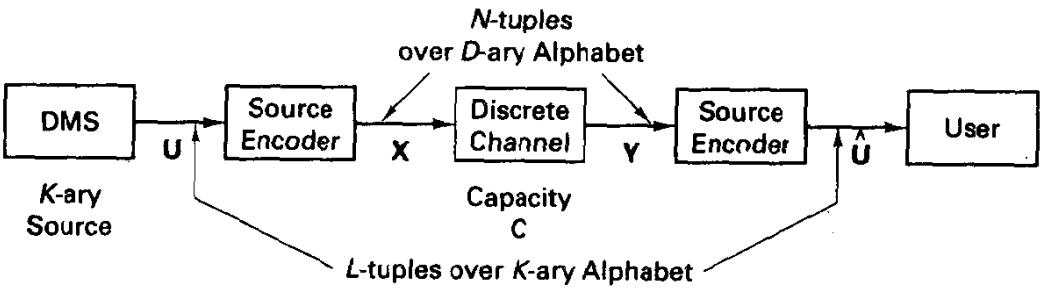


Figure 2.8.1 Block source coding diagram.

Under the assumption that we wish this encoding/decoding to be perfect, we must have a one-to-one relation between source vectors and codewords, which requires $K^L \leq D^N$, or that the codeword lengths be governed by

$$\frac{N}{L} \geq \frac{\log K}{\log D}. \tag{2.8.2}$$

Example 2.30 Binary-coded-decimal (BCD) Encoding of Numerical Data

Decimal digits ($K = 10$) are to be represented one at a time by binary strings. The required code length by (2.8.2) is $N \geq \log_2 10 / \log_2 2 = 3.32$, so $N = 4$ binary digits suffices. The code is formed by assigning the binary expansion of the source digit; for example, 9 corresponds with 1001, and so on. This binary-coded-decimal scheme has a fair degree of

inefficiency since 16 source symbols could be represented by 4-bit codewords; to eliminate most of this inefficiency, since $10^3 < 2^{10}$ (but with near equality), we could encode $L = 3$ digits together and use binary codewords of length $N = 10$.

This type of block-to-block source coding is only efficient when the source is equiprobable or nearly so, that is, when $H(U) \approx \log K$. If the probabilities are not equiprobable or, more importantly, symbols are not independent, the entropy of the source can be much less than $\log K$, and in these cases more efficient coding procedures exist. To see how, we proceed to develop the notion of ***typical sequences.*** For a sequence $U_1, U_2, \ldots, U_L$ produced by a DMS, we recall that the self-information of the string $\mathbf{u}_i = (u_{i_1}, \ldots, u_{i_L})$ is

$$I(\mathbf{u}_i) = \log \frac{1}{P(\mathbf{u}_i)}. \tag{2.8.3}$$

By the memoryless property of the source, we have that

$$P(\mathbf{u}_i) = \prod_{t=1}^{L} P(u_{i_t}) \tag{2.8.4a}$$

and thus

$$I(\mathbf{u}_i) = \log \frac{1}{P(\mathbf{u}_i)} = \sum_{t=1}^{L} \log \frac{1}{P(u_{i_t})}. \tag{2.8.4b}$$

The self-information of the string is then the sum of self-informations of the respective symbols. We expect this sum to converge to L times the mean of each term in the sum, which is $H(U)$, by a law of large numbers. Indeed, the Chebyshev inequality of Section 2.3 would hold that

$$P(|\, I(\mathbf{u}) - LH(U) \,| \geq \epsilon) \leq \frac{\sigma^2}{L^2\epsilon^2}, \tag{2.8.5}$$

where σ^2 is the variance of the (random) self-information $I(U_t)$ of a single source symbol, assumed finite.

Accordingly, we define T_ϵ as the set of output sequences $\mathbf{u}$ for which the self-information is ϵ-close to the expected value; that is,

$$T_\epsilon = \left\{ \mathbf{u}_i : \left| \sum_{t=1}^{L} \log \frac{1}{P(u_{i_t})} - LH(U) \right| \leq \epsilon \right\}. \tag{2.8.6}$$

By (2.8.5), this ***typical set*** of sequences holds probability at least $1 - \sigma^2/L^2\epsilon^2$, approaching 1 as L increases. Furthermore, if we invoke the relation between self-information and sequence probability, (2.8.4b), we have that for all sequences in T_ϵ

$$2^{-LH(U)-\epsilon} \leq P(\mathbf{u}_i) \leq 2^{-LH(U)+\epsilon}. \tag{2.8.7}$$

This result says that all sequences in the typical set have virtually equal (and small) probability, increasingly true for large L, and is referred to as the ***asymptotic equipartition property.***

Since the typical set has probability at most 1, we can upper-bound the number of sequences in the typical set by

$$|\, T_\epsilon \,| \leq 2^{LH(U)+\epsilon}. \tag{2.8.8}$$

If we are willing to allow a small probability ϵ of nonunique encoding, then for block source coding we need only assign codewords to $2^{LH(U)+\epsilon}$ typical source sequences. For D-ary coding of these sequences, this requires that $D^N \geq 2^{L[H(U)+\epsilon]}$ or that

$$\frac{N}{L} \geq \frac{H(U)+\epsilon}{\log D}, \tag{2.8.9}$$

which is an improvement over the ratio in (2.8.2). We have thereby demonstrated a source coding scheme whose number of code symbols per source symbol, N/L, is arbitrarily close to $H(U)/\log D$, and which has a probability of ambiguous encoding arbitrarily small. We might note the fundamental role of the law of large numbers here as well in attaining efficient encoding. Similar arguments lead to a converse theorem, to the effect that no block coding scheme can have N/L less than $H(U)$ while achieving arbitrarily small error probability.

Although clearly demonstrating the role of entropy in source coding, this block coding scheme has little practical significance due to the (small) probability of nonunique encoding. (Even sharpening the bound beyond Chebyshev inequalities leaves us with this nagging shortcoming.) Exercise 2.8.1 gives some numerical illustration of the problems to be encountered here.

2.8.2 Block-to Variable-length Encoding

Techniques that are practically important, however, are obtained by allowing either the source vector or codeword vectors, or perhaps both, to be of ***variable length.*** That is, either we collect a variable number of source symbols and associate with them fixed-length codewords, or vice versa, or allow both source and code strings to be of varying length. We shall consider only the fixed length-to-variable length techniques here, and develop a popular variable length-to-variable length scheme known as run-length encoding in the exercises.

Our aim previously was to minimize the (fixed) codeword length N. Now with codewords of varying lengths, $N_i, i = 1, 2, \ldots, K^L$, assigned to the possible source L-tuples, we attempt to minimize the average or expected codeword length, denoted $\overline{N}$. For an ergodic source, this would minimize the time-averaged number of code symbols expended per source symbol.

The additional difficulty that we encounter in this case is that of correctly parsing the received code stream, which is an arbitrary concatenation of codewords. By requiring the codewords to be ***prefix-free***, that is, insisting that no codeword be a prefix of any other, then the decoder can certainly parse the received string, for as soon as the decoder discovers a valid codeword in the string, it may terminate that codeword, knowing that the string is not the prefix of a longer codeword. (We will not discuss the practical issue of start-up in the middle of a message and/or recovery from channel errors.)

In describing prefix-free codes, it is convenient to associate codewords with nodes in a D-ary tree, where the actual codeword is given by the sequence of D-ary branching decisions to reach a given node. The prefix-free property requires simply that no node in the tree that is assigned to be a codeword can have parent nodes at subsequent levels that are also codewords. Figure 2.8.2a illustrates a possible binary ($D = 2$) prefix-free tree for four codewords, and Figure 2.8.2b shows another prefix-free assignment of

four codewords. Figure 2.8.2c, however, does not represent a prefix-free code. (The decoder for this code can nonetheless unambiguously decode, for the code symbol 0 is a start-of-codeword symbol.)

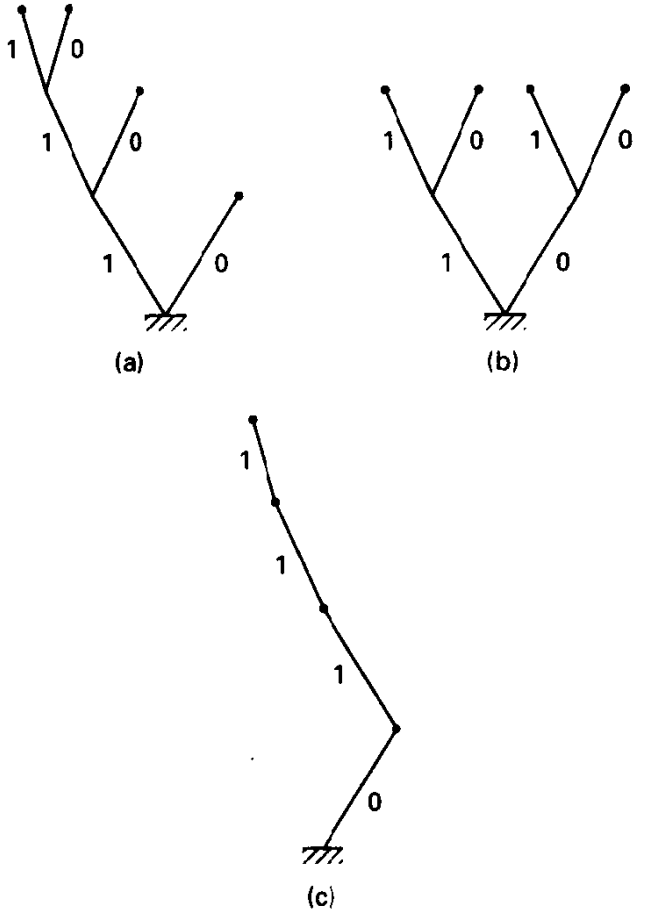


Figure 2.8.2 Binary variable length code trees, 4 codewords in each case, $K = 2$, $L = 2$, $D = 2$. (a) $K = 2$ prefix-free code tree; (b) $K = 2$ prefix code tree; (c) uniquely decodable, but not prefix-free code tree.

We next argue that for a prefix code the set of possible codeword lengths N_i, $i = 1, \ldots, K^L$, must satisfy an inequality due to Kraft [15]:

$$\sum_{i=1}^{K^L} D^{-N_i} \leq 1. \tag{2.8.10}$$

Conversely, we will see that when (2.8.10) is satisfied, we can construct a D-ary prefix-free code for the source (McMillan [16]).

To prove these claims, we first consider any D-ary prefix-free code and order the proposed codeword lengths in nondecreasing order, $N_1 \leq N_2 \leq \cdots \leq N_{K^L}$. Pick any node at depth N_1 in a D-ary tree as a codeword. This eliminates a fraction D^{-N_1} of nodes from an infinite tree. Assign another node at level N_2 in the remaining tree as the second codeword, which removes another fraction D^{-N_2} of the tree. When we encounter the coding assignment for the K^L-th codeword, we require at least one node remaining in the tree, and thus the Kraft inequality is *necessary* for prefix-free codes.

To prove *sufficiency* of the inequality (2.8.10), we observe that if a set of codeword lengths meets the Kraft inequality, then we may choose as our first codeword any node at depth N_1, the second codeword as any node not in the subtree of the first codeword (so that the prefix property is maintained), and so on, allowing us to construct a code with these lengths.

In addition to providing a test for the existence of prefix codes of specified lengths (see Exercise 2.8.2), the Kraft inequality allows us to prove that no prefix-free source coding scheme can have an average length $\overline{N}$ less that $H(X)/\log D$, as shown next.

We now prove the ***variable-length source coding theorem*** for discrete memoryless sources and a converse, which firmly install entropy as the source parameter of interest in coding of discrete sources.

Theorem. For a discrete memoryless source with entropy $H(U)$, a prefix-free code for source sequences of length L exists with expected number of code symbols per source symbol, $\overline{N}$, given by

$$\overline{N} < \frac{H(X)}{\log D} + \frac{1}{L}, \tag{2.8.11}$$

where D is the alphabet size of the codewords. Conversely, any prefix-free code must have

$$\overline{N} \geq \frac{H(X)}{\log D}. \tag{2.8.12}$$

Proof for $L = 1$ Case. The lower bound statement, (2.8.12), can be proved by showing that $H(X) - \overline{N} \log D \leq 0$ for any prefix-free code:

$$\begin{aligned} H(X) - \overline{N} \log D &= \sum_i P_i \log \frac{1}{P_i} - \sum_i P_i N_i \log D \\ &= \sum_i P_i \log \left(\frac{D^{-N_i}}{P_i} \right) \end{aligned} \tag{2.8.13}$$

Using the information theory inequality, $\log z \leq (z-1)/\log_e 2$, we find

$$H(X) - \overline{N} \log D \leq \frac{\sum_i D^{-N_i} - \sum P_i}{\log_e 2}. \tag{2.8.14}$$

However, the second sum is 1, and the Kraft inequality requires the first sum to be less than or equal to 1, verifying the necessity of (2.8.12).

To demonstrate the positive side, (2.8.11), suppose we choose the codeword lengths in accordance with the source output probabilities:

$$D^{-N_i} \leq P_i \leq D^{-(N_i - 1)}. \tag{2.8.15}$$

We can indeed create a prefix code having these lengths since $\sum D^{-N_i} \leq \sum P_i = 1$, and the Kraft inequality is satisfied. Taking logarithms of the right-hand portion of (2.8.15) gives

$$\log P_i \leq -(N_i - 1) \log D. \tag{2.8.16}$$

Multiplying both sides of (2.8.16) by P_i and summing yields

$$\sum_i P_i \log P_i \leq -\sum N_i P_i \log D + \sum P_i \log D. \tag{2.8.17}$$

Thus,

$$\overline{N} \leq \frac{H(U)}{\log D} + 1. \tag{2.8.18}$$

The proof may be easily extended to the case of encoding L source symbols at a time by noting that, for a DMS, the entropy of such L-tuples is just $LH(X)$, and then computing a bound on the mean codeword length for L-tuples and normalizing by L to obtain the average codeword length per source symbol.

Huffman coding

The preceding proof provides a construction of efficient source codes, referred to as Shannon–Fano coding, but this construction does not directly minimize the average codeword length. Huffman [17] developed an algorithm for the design of optimum block-to variable-length codes. We shall not prove the algorithm's optimality, but will merely discuss the procedure, beginning with the $D = 2$ case (coding with binary codewords).

We shall assume the task of encoding L-tuples obtained from a discrete source having K symbols in the alphabet and commence the construction by arranging the K^L source vectors, which we can regard as supersymbols, in descending order according to their probability. The two least probable are merged into a pseudosymbol having probability equal to the sum of the merged symbol probabilities. (These two codewords will ultimately differ in only the last position of the codeword.) Among the remaining $K^L - 1$ symbols, we combine the two least probable, adding their probabilities, and so on, until we have combined all symbols into one supersymbol (holding probability 1). This combining process is easily associated with a binary tree, and the codeword labeling corresponds to the sequence of combining decisions. An example will illustrate the algorithm.

Example 2.31 Huffman Coding for 8-ary Source, L = 1

Recall the earlier Example 2.10 pertaining to quantizing a Gaussian random variable. The optimal eight-level uniform quantizer may be shown to have zone probabilities of Gaussian p.d.f. between thresholds specified by Max [6]. Assuming the Gaussian sequence is an independent sequence, the quantizer output is an 8-ary DMS. We wish to encode this quantized source with binary codewords.

The Huffman code tree is shown in Figure 2.8.3, where the remaining two least probable symbols are combined at each stage. Binary codewords are obtained by reading backward in the tree from the root node. The expected codeword length for this code is, referring to Figure 2.8.3,

$$\begin{aligned}\overline{N} &= 2(0.23 + 0.23) + 3(0.17 + 0.17 + 0.08) + 4(0.08) + 5(0.02 + 0.02) \\ &= 2.70 \text{ bits per source symbol.}\end{aligned} \tag{2.8.19}$$

The source entropy $H(U)$ may be computed to be 2.65 bits/source symbol, so further gains in coding efficiency, which are available by coding multiple ($L > 1$) source symbols together, are small. Also, the gain over a standard fixed-length source code with 3 bits per codeword is a modest 12%, but the data compression can be much larger when either source symbol correlations exist or when the probabilities are more skewed.

For D-ary coding, with $D > 2$, the general procedure is quite similar: we usually group D symbols at each iteration, but wish to combine D at the last stage of the reduction process rather than the first. One way to ensure that this occurs is to add

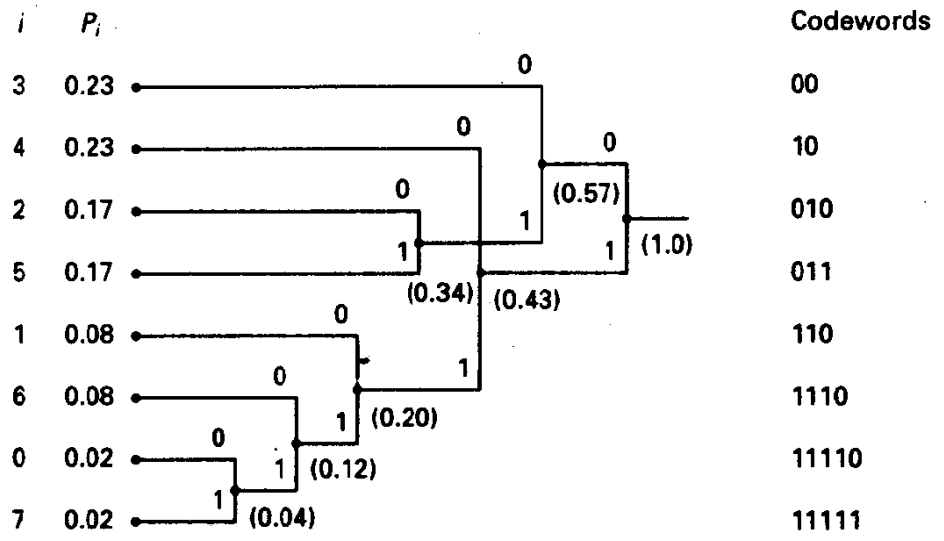


Figure 2.8.3 Huffman code tree for Example 2.31.

dummy symbols to the original set until $j(D-1)+1$ equals or exceeds the number of symbols for some integer j. Let the probabilities of these additional symbols be zero, and form the tree as usual. Obviously, these zero-probability symbols are never used, and they may be pruned from the code tree upon completion. Exercise 2.8.7 presents an application related to Example 2.31.

Huffman coding will achieve the lower bound of (2.8.12) in situations where the symbol probabilities are all inverse powers of D, for example, in a 4-ary source with symbol probabilities 0.5, 0.25, 0.125, and 0.125, a binary code is maximally efficient. Also, it is typical experience that the expected codeword length is much closer to the lower bound (2.8.12) than the upper bound (2.8.11). However, for some highly skewed sources, Huffman coding requires large values of L to achieve good efficiency, and procedures like ***run-length coding*** [18] are better suited (see Exercise 2.8.7). We note that this does not contradict the claimed optimality of Huffman codes; the latter are block- to variable-length codes, while run-length procedures are variable length-to-variable length encodings.

As a retrospective on source coding, we should grasp the uncertainty-reducing objective or, equivalently, the information-passing objective. The entropy of the source output is the same as that of the process of selecting a source codeword. After all, the two strings are merely two labelings of the same set of objects. If we want codewords to have short lengths on the average, then the amount of information conveyed, or uncertainty reduction, provided by each code symbol should be maximized. This will occur when the code tree is as nearly balanced as possible, meaning the routing in the tree is nearly equiprobable, given that we reach a given level. In Example 2.31, the first symbol of the codeword is a 0.57/0.43 binary random variable, with entropy near 1 bit. Given that the first symbol is a 1, the next symbol is a 0.535/0.465 binary variable, and so on.

A related analogy is the game of Twenty (or whatever number you wish) Questions, in which a friend thinks of an object in some predefined allowable set, such as household objects. Using a sequence of questions answerable by yes or no, you attempt to name the

object in 20 questions or less. If the allowable set included 1024 objects, but the friend tended to select them with unequal probabilities so that the entropy of the choice was only 7 bits, then a proper questioning strategy[21] would enable you to name the object in roughly 7 questions, on the average. Occasionally, you might require more than 10 questions, or you may be lucky and be successful on the first question. Of course, a friend who knows information theory will choose the objects equiprobably, implying that the best strategy will require 10 questions on average.

2.8.3 Extensions to Discrete Markov Sources

We shall conclude this section with remarks on discrete Markov sources, which often provide realistic models of sources with dependence. To define entropy for stationary sources with memory, we compute the entropy defined on L-tuples, that is, $H(\mathbf{U}_L)$, and then normalize by L to obtain the per-symbol entropy. For stationary sources, it may be shown that this ratio is monotone decreasing in L, and we define the entropy of the source as

$$H_\infty(U) = \lim_{L\to\infty} \frac{H(\mathbf{U}_L)}{L}. \tag{2.8.20}$$

For stationary Markov sources, this calculation is simplified as follows. Given that the system state at time k is $\sigma_k = j$, the source entropy is

$$H(U \mid \sigma_k = j) = -\sum_{i=1}^{K} P(u_i \mid \sigma_k = j) \log P(u_i \mid \sigma_k = j). \tag{2.8.21}$$

This will in general differ from state to state. The entropy for an ergodic Markov source is then given by the weighted sum of these conditional entropies:

$$H(U) = \sum_{j=0}^{S-1} P(\sigma = j) H(U \mid \sigma = j), \tag{2.8.22}$$

where $P(\sigma = j)$ is the steady-state probability of occupying state j. This result is intuitively expected, but is rigorously demonstrated in [12].

Example 2.32 Entropy of a Ternary Source with Memory

Suppose a ternary source has the model shown in Figure 2.8.4a. The input sequence W_k is an independent, equiprobable sequence drawn from {0, 1}, and this sequence drives a first-order recursive system defined by

$$U_k = U_{k-1} + W_k, \text{ modulo 3}, \qquad U_0 = 0. \tag{2.8.23}$$

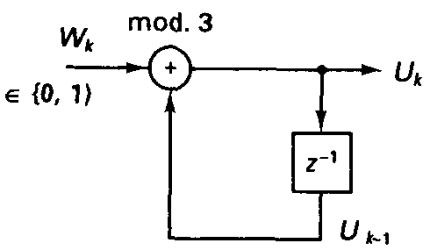


Figure 2.8.4a Source model.

[21] Assuming you knew the friend's p.m.f. for selecting objects.

We define the state at time k as $\sigma_k = U_k$, which takes on values in $\{0, 1, 2\}$. A state-transition diagram is shown in Figure 2.8.4b, from which symmetry makes it clear that the steady-state probabilities for all states are $\frac{1}{3}$. Furthermore, the entropy of the source, conditioned on any state, is 1 bit, because of the equiprobable chance of transiting to one of two next states. Thus, the source entropy is $H(U) = 1$ bit/symbol, somewhat less than the value of $\log_2 3 = 1.58$ bits/symbol obtained for a *memoryless* model with the same first-order statistics.

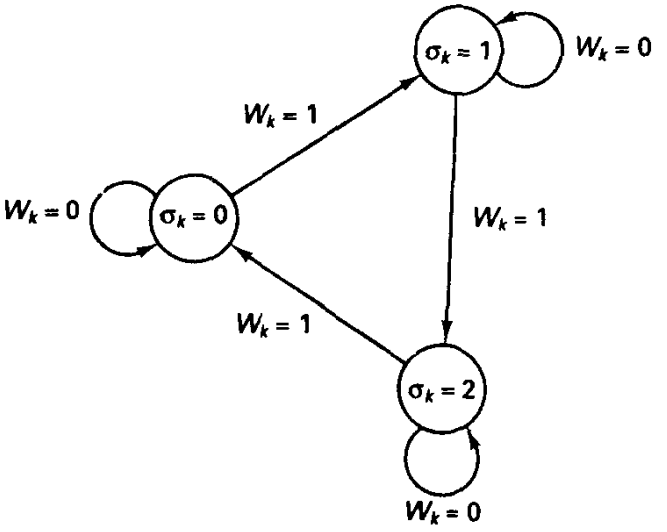


Figure 2.8.4b State transition diagram for Example 2.3.20.

Efficient coding of Markov sources can be accomplished through the use of state-dependent codes; that is, we develop a codebook for each state and track the state at both the encoder and decoder to correctly encode and decode. In Example 2.32, such a codebook is a trivial codebook with 1 bit per source symbol, conveying for each state what the next state should be.

Design of source coders as described requires knowledge of the source probability structure, something which is often unavailable in practice or which may be time varying among several models. Consequently, there has been much attention given to ***universal source encoding*** schemes that attempt to efficiently code any discrete source in a class of sources. Perhaps the best-known scheme is due to Lempel and Ziv [19], and similar methods are routinely implemented for compression of text files (see Exercise 2.8.8). A compression factor of two seems readily achievable on text files, but graphical or numerical files give greatly different compressibilities. A wealth of information on text compression is found in [20].

2.9 INFORMATION THEORY FOR CONTINUOUS RANDOM VARIABLES AND PROCESSES

Our development in Section 2.7 for discrete ensembles carries over to the continuous, or mixed, random variable situation in rather straightforward fashion, with only minor care required to interpret the various quantities. This in turn leads to generalizations for sequences and to waveforms through the use of orthonormal series expansions.

2.9.1 Scalar Variable Case

First, let us assume X and Y are continuous r.v.'s with joint p.d.f. given by $f(x, y)$. By a partitioning, or quantizing, of the space, R^2, of the random variables, we can make the problem discrete, one we have treated in Section 2.7. Specifically, imagine a uniform rectangular tiling of the x–y plane, with tile size Δ by Δ, as depicted in Figure 2.9.1. Following the definition of Section 2.7, the average mutual information shared by the discretized random variables (X^Δ, Y^Δ) is, assuming small tile size,

$$I(X^\Delta; Y^\Delta) \approx \sum_i \sum_j f(x_i, y_j)\Delta x\, \Delta y \log\left[\frac{f(y_j \mid x_i)\Delta y}{f(y_j)\Delta y}\right], \tag{2.9.1}$$

where i and j index the partitions. Letting the partition size shrink toward zero, we obtain in the limit the integral expression

$$\begin{aligned} I(X; Y) &= \int\int f(x, y) \log\left[\frac{f(y \mid x)}{f(y)}\right] dy\, dx, \\ &= \int\int f(x, y) \log\left[\frac{f(x, y)}{f(x)f(y)}\right] dy\, dx, \end{aligned} \tag{2.9.2}$$

both similar to expressions developed in the discrete case, except probability densities replace probabilities and integrals replace sums.

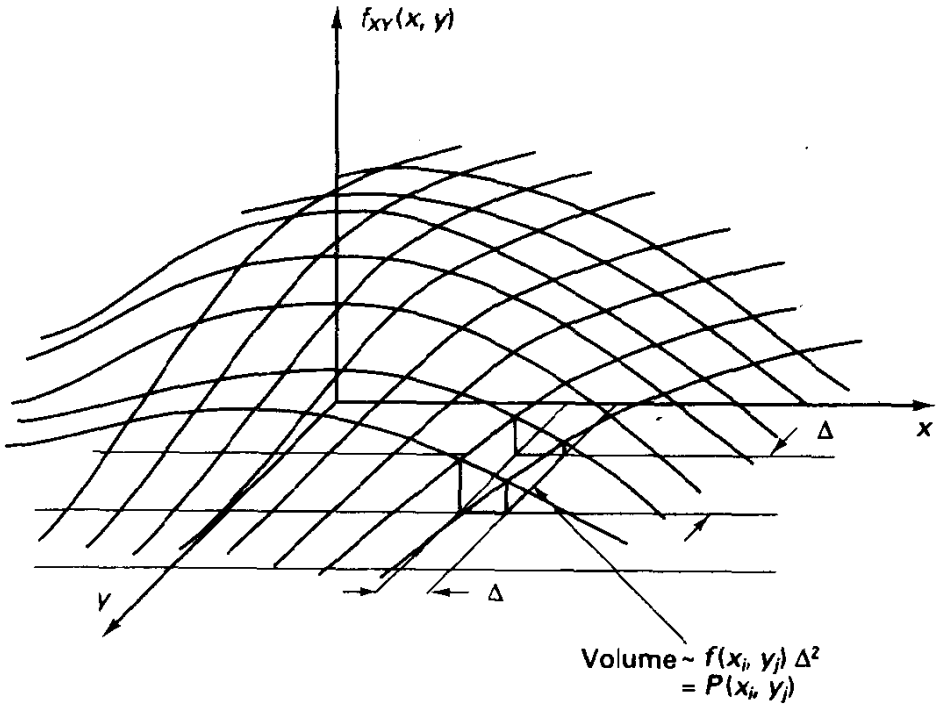


Figure 2.9.1 Discretization of continuous random variables, X, Y.

This might suggest that we define the entropy for the random variable Y as

$$H(Y) = -\int f(y) \log f(y)\, dy \tag{2.9.3}$$

and the conditional entropy for Y, given X, as

$$H(Y \mid X) = -\iint f(x, y) \log f(y \mid x)\, dy\, dx, \tag{2.9.4}$$

which would allow us to write

$$I(X; Y) = H(Y) - H(Y \mid X), \tag{2.9.5}$$

as in the discrete case. There is no conceptual difficulty with the definition of mutual information as in (2.9.2), and the same properties hold for it as were previously obtained for the discrete case. Entropies, however, require some care; in particular $H(Y)$ and $H(Y \mid X)$ are not limits of entropy quantities for the discretized problem as the tile size diminishes to zero, but are entropies relative to some common scale factor. Notice that if we arbitrarily scale the random variable X by some constant c, obtaining $X' = cX$, and adjust the p.d.f. for X' appropriately, we will find that $H(X')$ given by (2.9.3) differs from $H(X)$ by an amount $-\log(1/c)$. This is at obvious odds with our earlier interpretation of entropy as an uncertainty measure, for merely scaling the random variables seemingly has not changed the basic problem. We simply must forego the uncertainty interpretation in the continuous case, at least in the absolute sense,[22] noting that exact specification of the value of a continuous random variable requires an infinite number of yes/no questions anyway. Mutual information, however, as a difference of differential entropies, remains scale invariant.

The channel capacity for a continuous-input, continuous-output channel is defined as

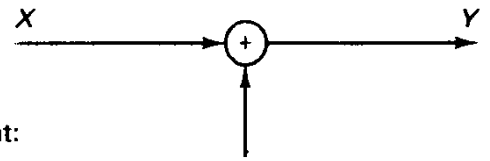
$$\begin{aligned} C &= \max_{f_X(x)} I(X; Y) \\ &= \max_{f_X(x)} \iint f_X(x) f(y \mid x) \log \left[\frac{f(y \mid x)}{\int f_X(z) f(y \mid z)\, dz} \right] dy\, dx, \end{aligned} \tag{2.9.6}$$

again similar in form to our earlier definition of channel capacity. In (2.9.6) we have written the mutual information of (2.9.2) in a form that explicitly shows the dependence on the input probability density $f_X(x)$. The variational problem is then to adjust the input probability density function $f_X(x)$ to maximize mutual information, perhaps subject to other constraints. [Technically speaking, the maximum need not exist in (2.9.6), and some treatments would therefore define C as the *supremum*, or *least upper bound*, on $I(X; Y)$. This need not concern us here, however.]

Example 2.33 Channel Capacity for Additive Gaussian Noise Channel

As an important case for our future study, consider the scalar Gaussian noise channel shown in Figure 2.9.2. The additive Gaussian noise Z is zero mean with variance σ^2 and is assumed independent of the input random variable X. To make the problem well posed, we place an average-energy, or mean-square-value, constraint on the input to the channel; that is, we insist that $\overline{X^2} \leq E$. Next, we note that the conditional (differential) entropy $H(Y \mid X)$ is

[22] In the continuous case, the entropies are usually termed differential entropies.



Constraint:

$\overline{X^2} \le E$ $\quad Z \sim \mathcal{N}(0, \sigma^2)$

Figure 2.9.2 Scalar additive Gaussian noise channel with energy constraint.

just the differential entropy, $H(Z)$, of the noise variable. This follows from substituting in (2.9.4) the fact that $f_{Y|X}(y \mid x) = f_Z(y - x)$ and integrating. Thus,

$$I(X; Y) = H(Y) - H(Z). \tag{2.9.7}$$

By direct calculation (using natural logarithms),

$$\begin{aligned} H(Z) &= -\int f(z) \log_e f(z)\, dz \\ &= -\int \frac{1}{(2\pi\sigma^2)^{1/2}} e^{-z^2/2\sigma^2} \left[\log_e \frac{1}{(2\pi\sigma^2)^{1/2}} - \frac{z^2}{2\sigma^2}\right] dz \\ &= \log_e \left[(2\pi\sigma^2)^{1/2}\right] + \frac{1}{2} = \log_e \left[(2\pi e\sigma^2)^{1/2}\right], \end{aligned} \tag{2.9.8}$$

which is invariant to the choice of the input probability density $f_X(x)$. Thus, to maximize mutual information, we must maximize $H(Y)$ subject to an input-energy constraint. Constraining the input energy to be less than or equal to E constrains the mean-square value of Y to be less than or equal to $E + \sigma^2$, so the problem becomes

$$\underset{f(y)}{\text{maximize}}\, H(Y) = -\int_{-\infty}^{\infty} f(y) \log_e f(y)\, dy \tag{2.9.9a}$$

subject to

$$\int_{-\infty}^{\infty} y^2 f(y)\, dy \le E + \sigma^2, \qquad \int_{-\infty}^{\infty} f(y)\, dy = 1. \tag{2.9.9b}$$

It is obvious that the maximum will be achieved when the input energy is the largest allowable, so we apply the method of Lagrange multipliers (see for example [21]) with an equality constraint. We form the objective function

$$\begin{aligned} G(f(y)) = &-\int_{-\infty}^{\infty} f(y) \log_e f(y)\, dy + \lambda_1 \left[\int y^2 f(y)\, dy - E - \sigma^2\right] \\ &+ \lambda_2 \left[\int f(y)\, dy - 1\right], \end{aligned} \tag{2.9.10}$$

where the λ_i are Lagrange multiplier constants. We are seeking to maximize $G(f(y))$ by adjusting $f(y)$ at all y. Differentiating with respect to $f(y)$ and setting the derivative to zero to obtain a local extremum, we determine after simplification that the optimizing p.d.f. is a Gaussian form,

$$f(y) = e^{-\lambda_1 y^2} e^{\lambda_2 - 1}, \tag{2.9.11}$$

where the Lagrange multipliers are chosen to satisfy the constraints. This in turn produces $\lambda_1 = -1/[2(E + \sigma^2)]$ and $\lambda_2 - 1 = \log_e [2\pi(E + \sigma^2)^{-1/2}]$. Thus, Y must be Gaussian with zero mean and variance $E + \sigma^2$, which can only be true if the input X is Gaussian with zero mean and variance E. Substituting the optimal output density into (2.9.9a) yields

$$H(Y) = \log_e [2\pi e(E + \sigma^2)^{1/2}] \text{ nats per channel use.} \tag{2.9.12}$$

The associated channel capacity is, by (2.9.7) and (2.9.8),

$$C = \frac{1}{2}\log_e\left(1 + \frac{E}{\sigma^2}\right) \text{ nats per channel use.} \tag{2.9.13}$$

Two ancillary aspects of this example, which can be demonstrated using similar methods, are as follows:

1. Under a variance constraint, the Gaussian distribution has the largest differential entropy.
2. For an additive noise channel with fixed noise variance σ^2 and an input energy constraint E, the Gaussian random variable is the worst-case choice for the noise distribution in terms of channel capacity.

Example 2.34 Capacity Calculation for a Gaussian Channel

Let $\sigma^2 = 10^{-6}v^2$ in a communication receiver, and, at the same point in the system, suppose we constrain the signal's second moment to be $E \leq 10^{-5}v^2$. The channel capacity, assuming Gaussian additive noise, is from (2.9.13)

$$C = \frac{1}{2}\log_e(1 + 10) = 1.199 \text{ nats per channel use}$$

$$= 1.730 \text{ bits per channel use.}$$

We interpret this as the maximum amount of mutual information between input and output variables under the preceding constraints, achievable if and only if the channel input is zero-mean Gaussian with the prescribed variance. Exercise 2.9.3 considers the maximum mutual information attainable when the input to this channel is four-level, meeting the energy constraint on the input, and when the output variable Y is quantized to four levels as well. This will give an idea of the sacrifice in capacity when a continuous-input, continuous-output channel is used with finite alphabets at each end.

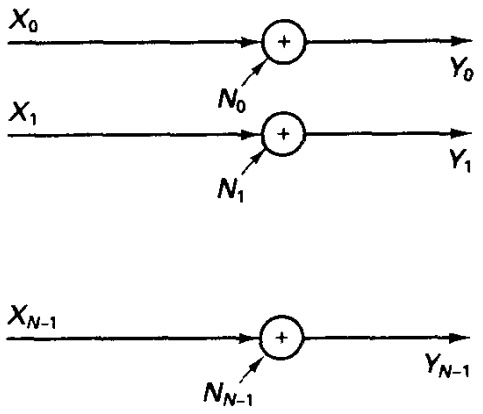
2.9.2 Vector Gaussian Channel Case

Next we consider vector channels, continuing with the Gaussian situation, as shown in Figure 2.9.3. Such a model may be obtained from successive transmissions in time or through the simultaneous use of several frequencies or antennas, and so on. We assume each additive noise channel is zero mean Gaussian with variance σ_n^2, $n = 0, 1, \ldots, N-1$, and that the noise variables are independent. A total energy constraint on the input vector is imposed, of the form $\sum \overline{X_i^2} \leq \sum E_i \leq E$, and we wish to find channel capacity.

First,

$$I(\mathbf{X}; \mathbf{Y}) = H(\mathbf{Y}) - H(\mathbf{Y} \mid \mathbf{X}) = H(\mathbf{Y}) - \sum_{n=0}^{N-1} \frac{1}{2}\log_e(2\pi e\sigma_n^2) \tag{2.9.14}$$

since the components of $\mathbf{Y}$ are independent, given the components of $\mathbf{X}$. Again our task is maximization of a differential entropy for a random vector subject to a second-moment



Constraint: $\sum \overline{X_i^2} \le E$

$N_i \sim \mathcal{N}(0, \sigma_i^2)$, Independent

Figure 2.9.3 Vector Gaussian channel with energy constraint.

constraint. This differential entropy is largest when the components of **Y** are independent and Gaussian, implying that the input components in **X** are independent Gaussian variables as well. The task is then a resource allocation problem:

$$\text{maximize} \sum_{i=0}^{N-1} \log\left[2\pi e(E_i + \sigma_i^2)\right] \tag{2.9.15a}$$

subject to

$$\sum_{i=0}^{N-1} E_i = E. \tag{2.9.15b}$$

Methods of variational calculus show that the optimal energy distribution is expressed parametrically in terms of a parameter B by

$$\begin{aligned} E_i + \sigma_i^2 &= B, \qquad \text{wherever } B \ge \sigma_i^2, \\ E_i &= 0, \qquad \text{wherever } B < \sigma_i^2, \end{aligned} \tag{2.9.16}$$

and B is adjusted to satisfy the energy constraint. In other words, for those components of the input vector that are allocated *any* energy, the sum of this energy together with the noise variance must be constant. The optimal solution may be easily comprehended by visualizing a reservoir, with N flat-bottomed segments, each σ_n^2 above some reference level, as depicted in Figure 2.9.4. The reservoir bottom profile thus represents the noise variance profile. We fill the reservoir with fluid until we have used E units. The fluid seeks a constant level[23] and provides the optimal distribution of energy. More energy is allocated to those channels where noise is small, and some channels may not be utilized at all if the total energy allocation is insufficient.

The capacity resulting from this optimal allocation is

$$C = \sum_{n=0}^{N-1} \frac{1}{2} \log_e \left(1 + \frac{E_n}{\sigma_n^2}\right) \text{ nats/vector channel use.} \tag{2.9.17}$$

[23]We provide a means for fluid to flow from one valley to another.

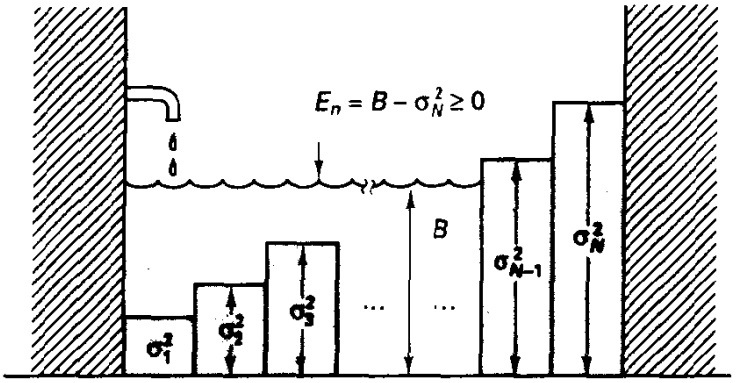


Figure 2.9.4 Reservoir analogy for optimal energy distribution on vector Gaussian channel.

Example 2.35 Two Gaussian Channels in Parallel

Consider the communication channel shown in Figure 2.9.5 on page 120, wherein we have two channels, one with 10 times the noise variance of the other. If we allocate $E = 6$ units of energy, then we have from (2.9.16) that $E_0 + 1 = B$ and $E_1 + 10 = B$, with B adjusted such that $E_0 + E_1 = 6$. Since energy in each channel must be nonnegative, the solution is to allocate all the energy to the less noisy channel. In contrast, if the total energy allocation is raised to 15 units, then the solution is $E_0 = 12$ and $E_1 = 3$. Note in particular that in the latter case the optimal distribution of energy is neither "all in one channel" nor allocated in the ratio of the noise variances for example. The joint density function for the optimizing input random variables is a bivariate Gaussian density, with independence between the two variables and with variances as given previously.

2.9.3 Waveform Channel Case

Consider the situation wherein the channel input is a continuous-time stationary random process $X(t)$, and the channel output is the process $Y(t) = X(t) + N(t)$, where $N(t)$ is a noise process, independent of $X(t)$. We inquire about the channel capacity for this situation, in units of bits per unit time.

The extension to the waveform channel case is provided through the concept of orthogonal series expansions, as discussed in Section 2.5. In particular, we argue that a T-second interval of a random process may be equated with a vector of expansion coefficients and that with proper choice of basis functions (the K–L basis), these expansion coefficients are uncorrelated random variables. When the process in question is Gaussian, the expansion coefficients are Gaussian as well and independent. In this framework, the parallel Gaussian channel just studied provides expressions for channel capacity.

To find capacity for the waveform channel case, we define C_T as the capacity in bits per T-second interval. This we will find through the vector representations given previously. The channel capacity in units per second is then defined as

$$C = \lim_{T \to \infty} \frac{C_T}{T} \text{ bits/second.} \tag{2.9.18}$$

We illustrate this process with the classical case of an ideal band-limited Gaussian noise channel.

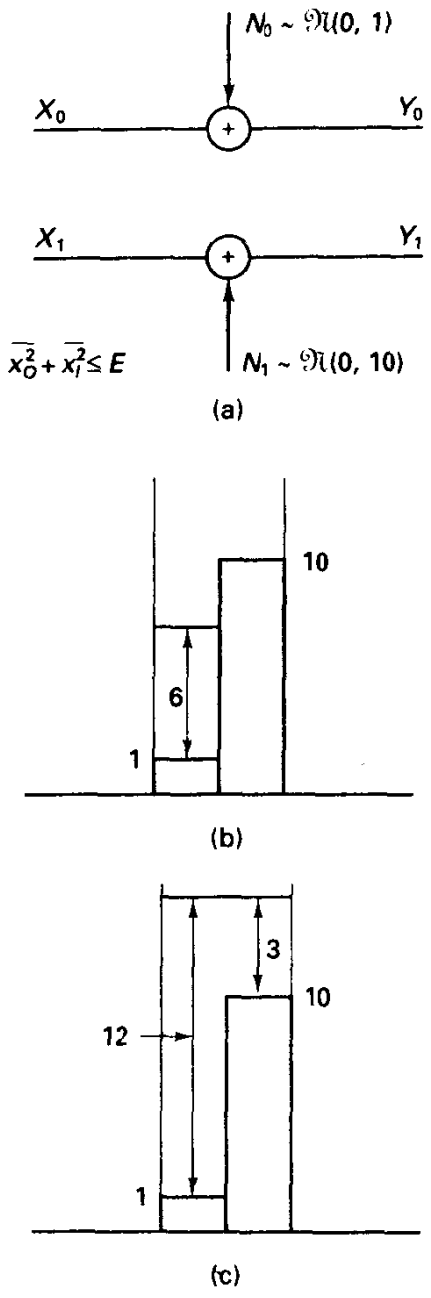


Figure 2.9.5 Parallel Gaussian channel with solutions for energy distribution. (a) Channel model; (b) $E = 6$ units; (c) $E = 15$ units.

Example 2.36 Capacity of Time-continuous Ideal Band-limited Gaussian Noise Channel

Consider the model of Figure 2.9.6a, where by including an ideal low-pass filter we force the transmitted signal $Z(t)$ to be strictly band-limited to B hertz. We place an average power constraint on $Z(t)$, that is, $E[Z^2(t)] \leq P$ watts. To this signal we add white Gaussian noise $N(t)$ with spectral density $N_0/2$ watts/hertz. We first consider calculating C_T by treating a T-second interval for both processes. The power constraint means that the available energy in the signal $Z(t)$ is PT.

As discussed in Section 2.5, the Karhunen–Loeve expansion provides an association between a Gaussian random process, say $N(t)$, and a random vector of expansion coefficients

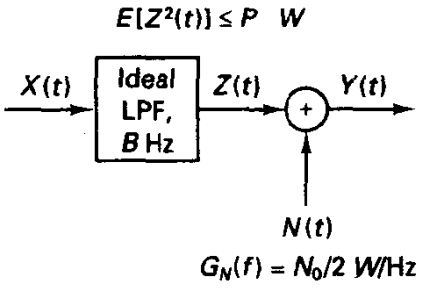


Figure 2.9.6a System model for bandwidth-limited communication on Gaussian noise channel.

whose components are independent, zero-mean, Gaussian random variables. As we have discussed, to expand $N(t)$, any orthonormal set of basis functions leaves noise variates that are independent and Gaussian, with variance $N_0/2$. For the ideal band-limited process $Z(t)$, however, special basis functions were necessary to produce independent, Gaussian expansion coefficients. Thus, we adopt for the basis set the prolate spheroidal functions discussed in Section 2.5 and thereby produce the infinite-dimensional parallel channel rendition shown in Figure 2.9.6b. The expansion is energy preserving in the sense that $\sum E[Z_i^2] = \sum \lambda_i = PT$, and the profile shown in Figure 2.9.6c shows that adopting the K–L expansion has inherently tailored the channel energy distribution automatically according to the eigenvalues of the K–L expansion.

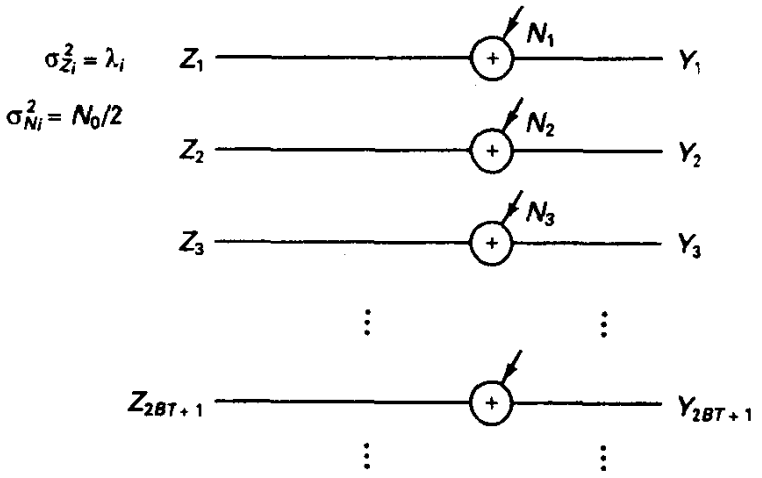


Figure 2.9.6b Parallel channel model via K–L decomposition.

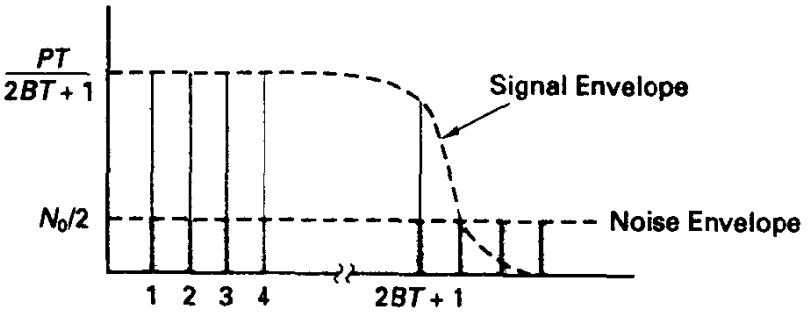


Figure 2.9.6c Profiles of signal and noise energy versus channel number.

Now recall the *hardening* that sets in for any given B as T grows: there are roughly $2BT + 1$ eigenvalues that are significant, each with value approximately $PT/(2BT + 1)$, and the remaining eigenvalues are negligibly small. Thus, for T large, the parallel channel capacity is

$$C_T \approx \sum_{i=1}^{2BT+1} \frac{1}{2} \log \left[1 + \frac{\lambda_i}{(N_0/2)} \right], \tag{2.9.19}$$

where $\lambda_i \approx PT/(2BT + 1)$.

We now formulate the channel capacity for the ideal band-limited Gaussian channel C_B according to (2.9.18) as

$$\begin{aligned} C_B &= \lim_{T \to \infty} \frac{C_T}{T} = \lim_{T \to \infty} \frac{2BT + 1}{2T} \log \left[1 + \frac{2PT}{(2BT + 1)N_0} \right] \\ &= B \log_e \left(1 + \frac{P}{N_0 B} \right) \text{ nats per second.} \end{aligned} \tag{2.9.20}$$

This is a classical expression due to Shannon for the ***capacity of the ideal band-limited Gaussian noise channel*** and is often expressed in terms of available signal-to-noise ratio $S/N = P/N_0 B$ as

$$C_B = B \log_e \left(1 + \frac{S}{N} \right) \text{ nats per second.} \tag{2.9.21}$$

(This expression holds also for ideal *bandpass* Gaussian noise channels where B is the channel bandwidth.)

An important special case of this result arises when the allowed bandwidth of the channel becomes arbitrarily large, but we operate with fixed power P watts and noise spectral density $N_0/2$ watts/hertz. Using (2.9.21) and the approximation that $\log_e(1 + x) \approx x$ as x becomes small, we obtain

$$C_\infty = \frac{P}{N_0} \text{ nats per second} = 0.693 \left(\frac{P}{N_0} \right) \text{ bits per second.} \tag{2.9.22}$$

In practical terms, the infinite-bandwidth capacity is essentially obtained when we use bandwidth B such that $P/N_0 B \leq 0.2$. Notice also that having small signal-to-noise ratio measured in the available bandwidth is not inherently to be avoided.

Another means of interpreting (2.9.21) is to ask, "What S/N is required for a system to supply C bits/second of channel capacity while operating within a bandwidth of B hertz?" Solving for S/N in (2.9.21) requires that this critical S/N be

$$\frac{S}{N} = \frac{P}{N_0 B} = 2^{C/B} - 1, \tag{2.9.23}$$

where we measure C in bits per second.

Important implications for digital communication follow from this capacity expression. Suppose we wish to transmit binary information at a rate of R_b bits per second, and we model the bits as independent. Thus, the source has entropy $H(U) = R_b$ bits/second. We must perform the transmission within channel bandwidth B hertz in the presence of additive Gaussian noise with spectral density $N_0/2$ watts/hertz. The available signal power at the receiver is P watts.

A generalization of the earlier converse to the coding theorem, proved for DMCs, would hold that for reliable transmission to be *possible* we must have $R_b < C$, both in equal units. Thus, from (2.9.20) we require that

$$R_b < B \log_2 \left(1 + \frac{P}{N_0 B} \right) = B \log_2 \left(1 + \frac{E_b R_b}{N_0 B} \right), \tag{2.9.24}$$

where the energy per bit E_b is just the received power P multiplied by the bit duration T_b and $R_b = 1/T_b$.

We will see in Chapter 3 that E_b/N_0 is a standard figure of merit for digital communication on the Gaussian noise channel. Solving in (2.9.24) for the minimum allowable E_b/N_0, we have the requirement for reliable transmission on the band-limited Gaussian noise channel that

$$\frac{E_b}{N_0} > \frac{B}{R_b}\left(2^{R_b/B} - 1\right). \tag{2.9.25}$$

Thus, the critical E_b/N_0 figure of merit is related only to the ***available bandwidth expansion ratio***, B/R_b. Figure 2.9.7 presents this lower bound versus B/R_b and reveals two important features. First, if essentially unlimited bandwidth is available, we must supply at least $E_b/N_0 = 0.693 = -1.6$ decibels to have any hope of reliable communication. (We have not yet demonstrated that in principle it is possible to communicate reliably with just slightly larger signal-to-noise ratio.) Furthermore, when bandwidth is constrained, the required energy-to-noise density ratio increases dramatically, essentially in exponential fashion, when the ratio R_b/B exceeds unity. This topic will be further developed in Chapter 3, where we compare typical modulation schemes against this standard, and in our later discussion of coding methods.

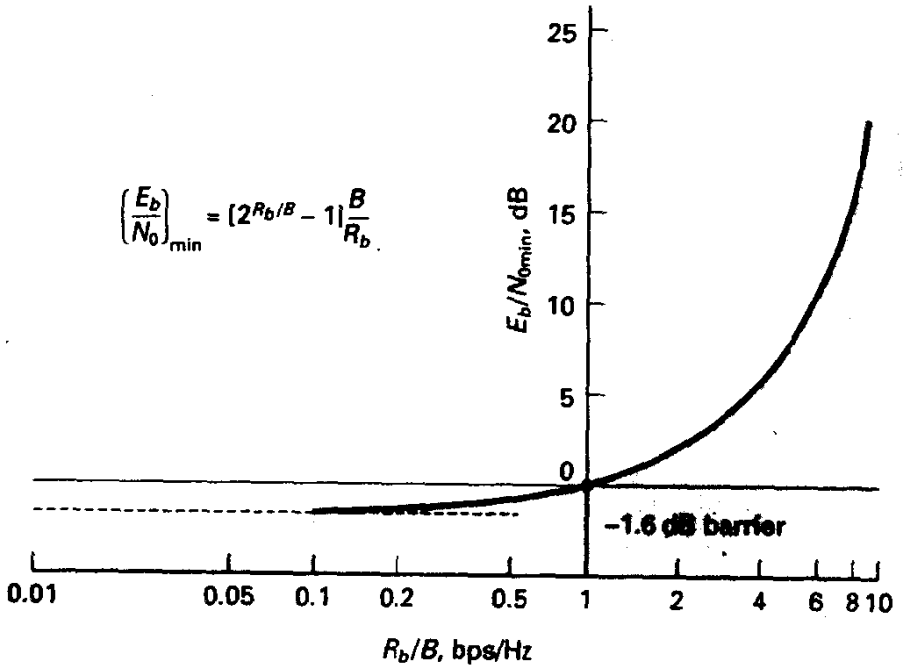


Figure 2.9.7 Channel capacity limitation for digital communication on band-limited additive Gaussian noise channel.

BIBLIOGRAPHY

1. Papoulis, A., *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, 1983.

2. Larson, H. J., and Shubert, B. O., *Probabilistic Models in Engineering Sciences, Vols I and II*. New York: Wiley, 1979.

3. Gray, R. M., and Davisson, L. D., *Random Processes: A Mathematical Approach for Engineers*, Englewood Cliffs, NJ: Prentice Hall, 1986.

4. Leon-Garcia, A., *Probability and Random Processes for Electrical Engineering*. Reading, MA: Addison-Wesley, 1989.

5. Wozencraft, J. M., and Jacobs, I. M., *Principles of Communication Engineering*. New York: Wiley, 1965.

6. Max, J., "Quantizing for Minimum Distortion," IRE Transactions on Information Theory, pp. 7–12, March 1960.

7. Chernoff, H., "A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on a Sum of Observations," *Annals of Mathematical Statistics*, no. 23, pp. 493–507, 1952.

8. Loeve, M., *Probability Theory*. New York: Van Nostrand Reinhold, 1955. (Also found in [1].)

9. Slepian, D., and Landau, H. J., "Prolate Spheroidal Wave Functions, Fourier Analysis and Uncertainty, Part I," *Bell System Technical Journal*, vol. 40, pp. 43–64, 1961. (See also Parts II and III, Landau and Pollak, *Bell System Technical Journal*, 1961–1962.)

10. Gilbert, E. N., "Capacity of a Burst Noise Channel," *Bell System Technical Journal*, vol. 39, pp. 1253–1265, September, 1960; Elliot, E. O., "Estimates of Error Rates for Codes on Burst Noise Channels," *Bell System Technical Journal*, vol. 42, pp. 1977–1997, September, 1963.

11. Shannon, C. E., "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948, and vol. 27, pp. 623–656, October 1948.

12. Gallager, R. *Information Theory and Reliable Communication*. New York: Wiley, 1968.

13. Blahut, R., "Computation of Channel Capacity and Rate-distortion Functions," *IEEE Transactions on Information Theory*, IT-18, pp. 460–473, 1972.

14. Fano, R. M., *Transmission of Information*. Cambridge, MA: MIT Press, 1961.

15. Kraft, L. G., "A Device for Quantizing, Grouping, and Coding Amplitude Modulated Pulses," M.S. Thesis, MIT, Cambridge, MA, 1949.

16. McMillan, B., "Two Inequalities Implied by Unique Decipherability," *IRE Transactions on Information Theory*, vol. IT-2, pp. 115–116, 1956.

17. Huffman, D., "A Method for the Construction of Minimum Redundancy Codes," *Proceedings of IRE*, no. 40, pp. 1098–1101, 1962.

18. Meyer, H., Rosdolsky, G., and Huang, T., "Optimum Run Length Codes," *IEEE Transactions on Communications*, COM-22, no. 6, pp. 826–835, 1977.

19. Ziv, J., and Lempel, A., "Compression of Individual Sequences by Variable Rate Coding," *IEEE Transactions on Information Theory*, vol. IT-24, pp. 530–536, 1978; see also Welch, T., "A Technique for High-performance Data Compression," *IEEE Computer*, vol. 17, No. 6, 1984.

20. Bell, T. C., Cleary, J. G., and Whitten, I. H., *Text Compression*. Englewood Cliffs, NJ: Prentice Hall, 1990.

21. Blahut, R., *Principles and Practice of Information Theory*. Reading, MA: Addison-Wesley, 1987.

22. Shannon, C., "Prediction and Entropy of Printed English," *Bell System Technical Journal*, pp. 50–64, January 1951.

EXERCISES

2.1.1. Let A and B be two events in a field. Verify that $P(A \cup B) = P(A) + P(B) - P(A \cap B)$ by the use of simple set operations, the axioms of probability, and the possible aid of a Venn diagram.

2.1.2. A certain experiment has events A, B, and C with the following probabilities:

$$P(A) = \frac{1}{2}$$

$$P(B) = \frac{1}{3}$$

$$P(C) = \frac{1}{4}$$

$$P(A \cap B) = \frac{1}{6}$$

$$P(A \cap C) = \frac{1}{8}$$

$$P(B \cap C) = \frac{1}{12}$$

$$P(A \cap B \cap C) = \frac{1}{16}$$

(a) Are the events pairwise independent? Are the events jointly independent?
(b) Find $P(A \cup B)$.
(c) Find $P(A \cup B \cup C)$.
(d) Determine $P(B \mid C)$ and $P(A \mid B \cap C)$ using Bayes's rule.

2.1.3. In a binary communication system we have four messages 000, 110, 101, and 011. The messages are selected with equal probability, and sent through the binary symmetric channel of Example 2.3, which transmits each message symbol independently, with a probability of error equal to 0.1. Let $\mathbf{r} = (r_1, r_2, r_3)$ denote the channel output vector, and let B_i be events denoting that the received symbol r_i is 0.
(a) What is $P(B_1)$?
(b) What is the probability that $\mathbf{r} = (110)$?
(c) Given that we receive $\mathbf{r} = (110)$, what is the a posteriori probability that the message 110 was sent?
(d) Are the events B_i jointly independent? pairwise independent?

2.2.1. Derive the binomial probability mass function (2.2.3) for k successes in n independent trials of a binary experiment by determining the probability of a specific sequence of outcomes and then arguing that there are $C_k^n = n!/k!(n-k)!$ (disjoint) arrangements of k successes in n trials.

2.2.2. Suppose N is a Gaussian random variable representing receiver noise, having zero mean and variance 0.01 v^2. A constant signal added to N has amplitude 0.20 volt. Show that the probability that the sum is less than zero is $Q(2)$ and evaluate using (1) Table 2.1 of the Q-function and (2) the upper bounds on $Q(x)$ discussed in the text.

An approximation to $Q(x)$ that is very accurate for all arguments x is[24]

$$Q(x) \approx \left[\frac{1}{(1-a)x + a(x^2+b)^{1/2}}\right] \cdot \frac{1}{(2\pi)^{1/2}} e^{-x^2/2}$$

where $a = 0.344$ and $b = 5.334$. Compare this approximation with the two preceding results.

2.2.3. Two random variables X and Y have the joint p.d.f. given by

$$f_{XY}(x,y) = \begin{cases} B, & x^2 + y^2 \leq 1, \\ 0, & \text{else.} \end{cases}$$

(a) Find B to properly scale the density function, and sketch the joint p.d.f.
(b) Find the marginal p.d.f.'s for either X or Y.
(c) Determine and sketch the conditional p.d.f.'s $f_{Y|X}(y \mid x = 0)$ and $f_{Y|X}(y \mid x = 0.9)$.
(d) Are X and Y independent?

2.2.4. For the three-dimensional p.d.f. of Example 2.8, determine the marginal p.d.f. for X_1 and the conditional p.d.f. $f(x_2 \mid x_1)$. Show that both are of Gaussian form. (This is a general property associated with multivariate Gaussian r.v.'s.)

2.2.5. Let X be a Gaussian random variable with $\mu = 0$, and let $Y = \mid X \mid$.
(a) Use $F_Y(y) = P(Y \leq y) = P(\mid X \mid \leq y^{1/2})$ to find an integral expression for $F_Y(y)$. Then differentiate to find $f_Y(y)$.
(b) Use the result of part (a) to find the mean of Y.

2.2.6. Uniform random variables are often provided as system subroutines on most computers. If none is available, Leon-Garcia [4] suggests the following algorithm for producing uniformly distributed random numbers on a computer: let Z_i be defined recursively by

$$Z_i = 7^5 Z_{i-1} \bmod (2^{31} - 1)$$

with Z_0 being the "seed" of the sequence. The sequence of integers nas a period of $2^{31} - 1$. To obtain values uniform on [0,1], simply normalize Z_i after the recursion by 2^{31}. Write a short program to generate 1000 variates, and compute a crude histogram by counting the number in each decile of the range.

2.2.7. The Box–Muller method is a popular method of generating exactly Gaussian random variables on a computer, given the availability of a uniform random number generator. Let U_1 and U_2 be uniformly distributed on [0,1] and independent. Then

$$X = (-2\log_e U_1)^{1/2} \cos(2\pi U_2),$$

$$Y = (-2\log_e U_1)^{1/2} \sin(2\pi U_2)$$

are independent, zero-mean, unity variance Gaussian random variables. (Showing this result is a superb exercise in the transformation of random variables [1, Chapter 6].) Write a short program to generate 1000 such variables and test the sample mean, sample variance, and sample correlation between X and Y to support at least part of the preceding claim.

2.3.1. The geometric random variable has probability mass function given by (2.2.5). Show that the mean of the waiting time W until, and including, the time when the next error occurs

[24] P. O. Borjesson and C.-E. W. Sundberg, "Simple Approximations of the Error Function Q(x) for Communications Applications," *IEEE Transactions on Communications*, vol. COM-27, pp. 639–643, March 1979.

in a sequence of independent transmissions, each with error probability ϵ, is

$$E[W] = \sum_{j=1}^{\infty} j\epsilon(1-\epsilon)^{j-1} = \frac{1}{\epsilon}.$$

Thus, the mean waiting time until the next error is simply the inverse of the error probability. *Trick:* Realize the expression for the mean is the infinite sum $\sum_{k=0}^{\infty} p(d/dq)(q^k)$, where $p = \epsilon$ and $q = 1 - p$. Exchange differentiation and summation, which is permissible since the infinite series is convergent.

2.3.2. Verify by direct integration that the standard deviations of the uniform and Gaussian random variables of Examples 2.5 and 2.6 are $(b-a)/12^{1/2}$ and σ, respectively. A table of definite integrals may be useful in the Gaussian case.

2.3.3. For the random variables X and Y described in Exercise 2.2.3:

(a) Show that X and Y are uncorrelated.

(b) Show that X and Y are not independent, thus refuting a possible claim that uncorrelatedness implies independence.

2.3.4. If Θ is uniformly distributed on the interval $[0, 2\pi]$, show by definition of the expectation operator that

(a) $E[\cos\Theta] = 0$

(b) $E[\cos^2\Theta] = \frac{1}{2}$

(c) $E[\cos\Theta \sin\Theta] = 0$

2.3.5. Let $\mathbf{X}$ be an n-dimensional Gaussian vector of r.v.'s with mean vector $\mathbf{m}_x$ and covariance matrix $\mathbf{K}_x$. Let $\mathbf{Y} = \mathbf{AX} + \mathbf{b}$ be a linear transformation of the original vector, where $\mathbf{A}$ is a $n \times n$ matrix and $\mathbf{b}$ is $1 \times n$ vector.

(a) Show from the definitions that the mean vector and covariance matrix of $\mathbf{Y}$ are $\mathbf{m}_y = \mathbf{Am}_x + \mathbf{b}$ and $\mathbf{K}_y = \mathbf{AK}_x\mathbf{A}^T$, respectively, where T denotes the matrix transpose operation.

(b) Show, furthermore, that the p.d.f. for $\mathbf{Y}$ is of Gaussian form by solving for $\mathbf{X}$ in terms of $\mathbf{Y}$ (assume $\mathbf{A}$ is invertible) and substituting into the general Gaussian form.

2.3.6. The Cauchy random variable has p.d.f. described by

$$f_X(x) = \frac{1}{\pi(1+x^2)}, \qquad -\infty < x < \infty.$$

(a) Determine the characteristic function by computing the Fourier transform of this p.d.f.

(b) Use this result to show that the sum of N independent Cauchy r.v.'s is also Cauchy distributed, but with a different scale parameter. Thus, like the Gaussian random variable, the Cauchy variable is "reproducing" under summation.

(c) Does the mean, or any moment for that matter, of this random variable exist? By symmetry, we are tempted to say the mean is zero, but check the definition of mean.

2.3.7. Jensen's inequality holds that if $f(x)$ is a $\cap$ function of a random variable X, then

$$E[f(X)] \leq f(E[X]) \tag{1}$$

or, in words, the function evaluated at the mean of X is at least as large as the expected value of the r.v. defined by $Y = f(X)$. If the function is $\cup$, then the inequality is reversed.

Proof. We sketch the proof for the case of a discrete r.v. having N values. Extension to countably infinite discrete cases, or continuous r.v. cases, follows the same argument.

First consider the case $N = 2$; that is, X is binary r.v. taking on values x_1 and x_2, with probabilities p_1 and p_2, summing to 1. The left-hand side of Jensen's inequality is $p_1 f(x_1) + p_2 f(x_2)$, which can be interpreted as a point on the line joining $[x_1, f(x_1)]$ and $[x_2, f(x_2)]$, as shown in Figure P2.3.7. But if $f(x)$ is convex $\cap$, this value is never larger than $f(p_1 x_1 + p_2 x_2)$, which is the right-hand side in (1). Thus, we have demonstrated that the inequality holds for the binary r.v. case.

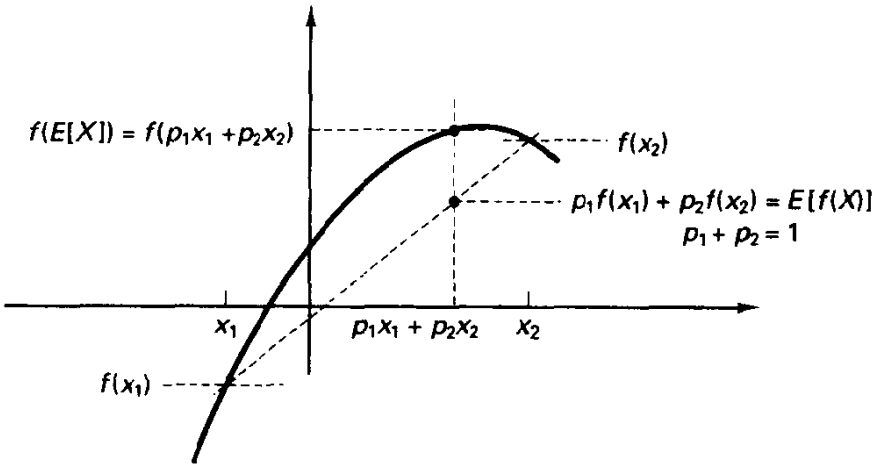


Figure P2.3.7 Interpretation of convexity and Jensen's inequality for binary r.v.

Now we use induction, assuming that if the inequality holds for r.v.'s with $N - 1$ outcomes it then holds for r.v.'s with N outcomes. So, assume the former, and let the N outcomes be $x_1, x_2, \ldots, x_N$, with probabilities $p_1, p_2, \ldots, p_N$. The left-hand side in (1) is

$$\begin{aligned} \sum_{i=1}^{N} p_i f(x_i) &= \sum_{i=1}^{N-1} p_i \frac{\sum_{j=1}^{N-1} p_j f(x_j)}{\sum_{j=1}^{N-1} p_j} + p_N f(x_N) \\ &\leq \sum_{i=1}^{N-1} p_i f(\eta) + p_N f(x_N), \end{aligned} \tag{2}$$

where $\eta \equiv \sum_{j=1}^{N-1} p_j x_j / \sum_{j=1}^{N-1} p_j$, that is, the mean value of the first $N - 1$ outcomes, renormalized in probability. The inequality follows because of the assumption that Jensen's inequality holds for the $N - 1$ r.v.'s. Now we apply the inequality for the two-point r.v. having outcomes η and x_N, with probabilities $\sum_{j=1}^{N-1} p_j$ and p_N:

$$\sum_{i=1}^{N-1} p_i f(\eta) + p_N f(x_N) \leq f\left(\sum_{i=1}^{N-1} p_i \eta + p_N x_N\right). \tag{3}$$

Combining (2) and (3), we obtain

$$\sum_{i=1}^{N} p_i f(x_i) \leq f\left(\sum_{i=1}^{N} p_i x_i\right),$$

which is the desired inequality (1). Since the inequality holds for N outcome variables if it holds for $N - 1$, and it also holds for $N = 2$, by induction the inequality holds for any r.v. with a finite number of outcomes.

2.3.8. Applications of Jensen's inequality

(a) Let $X \sim U[0, 1]$, and suppose that $f(x) = x^{1/2}$. Use Jensen's inequality to show that $E[X^{1/2}] \leq (1/2)^{1/2} = 0.707$. Find the actual p.d.f. for the r.v. $Y = X^{1/2}$ in this case; then determine the exact mean and compare.

(b) Suppose $X \sim U(0, 1)$ and that $g(x) = \log_e x$. Use Jensen's inequality to bound $E[\log_e |X|]$. *Hint:* First find the p.d.f. for the absolute value of X, which is always positive.

2.4.1. Let X be an exponential random variable; that is,

$$f_X(x) = \frac{1}{\lambda} e^{-x/\lambda}, \qquad x \geq 0.$$

(a) Find the cumulative distribution function $F_X(x)$.

(b) Determine the mean and variance of the r.v. X.

(c) Calculate the probability that X exceeds 3λ.

(d) Use the Markov inequality to bound the probability that X exceeds 3λ.

(e) Repeat, using the Chernoff bound, and compare.

2.4.2. Repeat the steps of Problem 2.4.1 for the Rayleigh random variable; that is,

$$f(x) = \frac{x}{\lambda^2} e^{-x^2/2\lambda^2}, \qquad x \geq 0.$$

2.4.3. Apply the Chebyshev inequality to the probability that four or more errors occur in the setting of Example 2.14.

2.4.4. Consider Example 2.14 and repeat the analysis for the event "8 errors in 200 trials" so that the relative success frequency is the same. Do you confirm a law of large numbers?

2.4.5. Give a precise statement of the weak law of large numbers applied to the transmission of N bits through a binary symmetric channel with error probability p, where successive uses of the channel are independent. In particular, how many errors n_e are expected with high confidence as N becomes large? Use the Chebyshev inequality to find N so that our empirical estimate of p, $\hat{p} = n_e/N$, is within 10% of the true p. This requires knowledge that the variance of the random variable $\hat{p}$ is $p(1 - p)/N$.

2.4.6. In evaluation of digital communication systems, simulation using random number generators is the usual procedure to measure empirically the system error rate. (This is particularly useful when the analysis becomes too intractable.) We can measure the time-averaged error probability by counting errors and dividing by the number of symbols processed through the system. Discuss the conceptual issues involved in equating such measurements with "probability of error" defined in the ensemble sense. Discuss how we should view the results of such measurements, that is, as a random variable itself that may (or may not) converge to the correct answer.

2.4.7. The central limit theorem would hold that if $X_1, X_2, \ldots, X_{10}$ are independent, zero-mean, unit-variance Gaussian variables, and if we form

$$Y = \sum_{i=1}^{10} X_i^2,$$

that the p.d.f. for Y should appear roughly Gaussian in form. (We might think that Y is exactly Gaussian, but it is obtained as a sum of *nonlinear* transformations of Gaussian variables.) We know in fact that the p.d.f. for Y is a chi-squared form given in (2.2.53). Plot the shape of this p.d.f. for $n = 10$ and $\sigma = 1$ and compare with a Gaussian p.d.f. having a mean of $10E[X^2] = 10$ and variance $10E[X^4] = 10 \cdot 3\sigma^4 = 30$.

2.5.1. Show that if $X(t)$ is a real, wide-sense stationary process, the following properties of the autocorrelation function and power spectral density hold:

(a) $R_X(\tau) = R_X(-\tau)$.

(b) The power spectrum $G_X(f)$ is real and even.

The first property follows from definition and shift of time origin, and property (b) follows from the result in part (a).

2.5.2. White Gaussian noise with spectral density $N_0/2$ watts/hertz is an input to the low-pass filter shown in Figure P2.5.2. For this circuit, the transfer function is

$$H(f) = \frac{1}{1 + j2\pi fRC}.$$

(a) Show the power spectrum of the output random process $Y(t)$ is

$$G_Y(f) = \frac{N_0}{2}\left[\frac{1}{1 + (2\pi fRC)^2}\right]$$

and by integration of this power spectrum that the mean-square value of $Y(t)$ is $\overline{Y^2(t)} = N_0/4RC$.

(b) By means of an inverse Fourier transform on $G_Y(f)$, show that

$$R_Y(\tau) = \frac{N_0}{4RC}e^{-|\tau|/RC}$$

(c) Discuss the effect of changing the *time constant* RC on the power spectrum and autocorrelation.

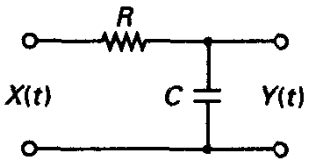


Figure P2

2.5.3. Formulate the bivariate p.d.f. for two samples of the random process $Y(t)$ described in Exercise 2.5.2. The bivariate Gaussian density function, a special case of (2.3.11), involves two means, two variances, and the correlation coefficient. Let the time constant $RC = 10^{-3}$ second, and consider two cases:

(a) Samples taken 0.1 millisecond apart, producing highly correlated r.v.'s

(b) Samples taken 10 milliseconds apart, producing essentially uncorrelated samples

In both cases, the level contours of equal probability density will be ellipses in the plane, centered at the origin.

2.5.4. The random binary wave was discussed in Example 2.18 and shown to have an autocorrelation function given in Figure 2.5.3b.

(a) Show that the power spectral density for the random process is

$$G_X(f) = T\frac{\sin^2(\pi fT)}{(\pi fT)^2}.$$

(b) Show that the power spectral density has a first null at $f = 1/T$.

(c) By use of integral tables, show that roughly 90% of the power of the signal is located in the frequency range $|f| \leq 1/T$.

2.5.5. Let X_m denote the expansion coefficient in the Karhunen–Loeve expansion for a zero-mean stationary random process. Show that $E[X_m] = 0$ and that $\text{Var}[X_m] = \lambda_m$, where λ_m is the eigenvalue attached to the mth expansion function. Show further that the power in the signal $X(t)$ may be expressed as $E[X^2(t)] = \sum_{m=0}^{\infty} \lambda_m$.

2.5.6. Set up but do not solve the integral equation whose solutions are the eigenfunctions for the K–L expansion of the random binary wave. Use an expansion interval of 10 bits and the known autocorrelation function for this random process. Using the $N = 2BT + 1$ rule, along with a bandwidth of the signal corresponding to twice its bit rate, determine the number of significant eigenvalues. Would the expansion coefficients be Gaussian random variables in this case?

2.5.7. Recall the definition of the alternate mark inversion technique depicted in Figure 1.1.2, except that let's define the pulses to be full-bit-width pulses of alternating polarity. Follow the same arguments used to analyze the random binary wave to show that the autocorrelation function is as shown in Figure P2.5.7 and that from this the power spectrum is of the form

$$G_X(f) = K \frac{\sin^4(\pi f T)}{(\pi f T)^2}.$$

Plot this spectrum, in particular showing that the spectral density has a null at zero frequency.

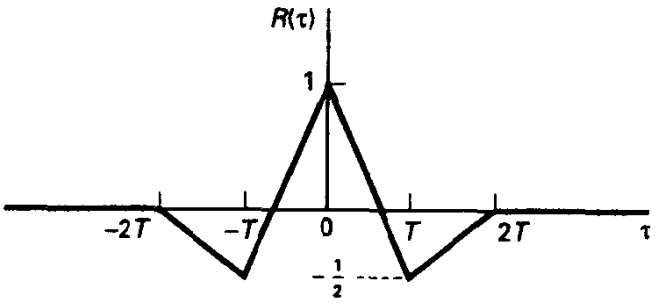


Figure P2.5.7

2.5.8. Suppose a white Gaussian discrete-time sequence is the input to the two discrete-time systems depicted in Figure P2.5.8. The first is a high-pass finite impulse response (FIR) filter, and the second is a low-pass infinite impulse response (IIR) filter. The respective z-transforms are

$$H_1(z) = 1 - z^{-1}$$

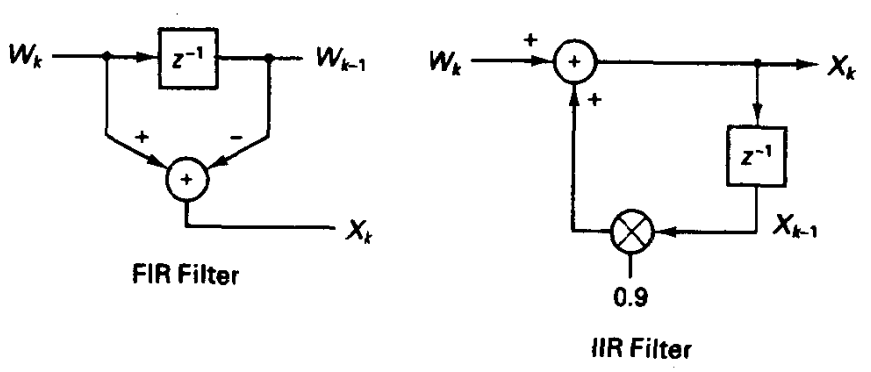


Figure P2.5.8

and

$$H_2(z) = \frac{1}{1 - 0.9z^{-1}}.$$

Using the fact that the input sequence has discrete-time power spectrum σ^2, that is, independent of frequency, use (2.5.35) and (2.5.36) to determine the output power spectrum in each case and the output autocorrelation sequence.

2.5.9. Consider the discrete-time system depicted in Figure P2.5.9, to reappear in Chapter 6. This is a finite-state Markov system, for which we define the state at time k to be $\sigma_k = (b_k, b_{k-1})$, so the system has four states. For each state transition, two output bits are produced as shown.

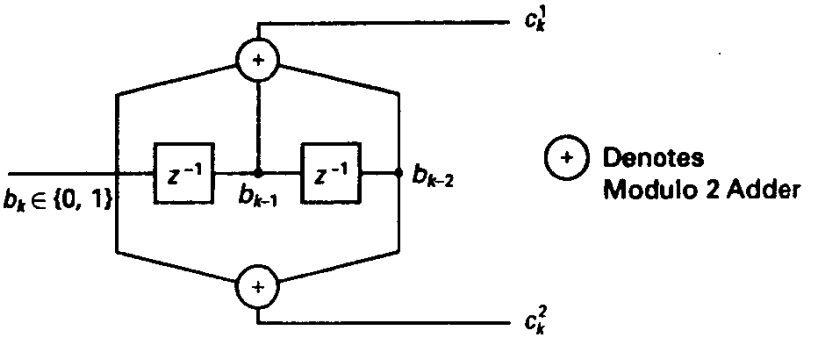


Figure P2.5.9

(a) Argue that any state has probability of transitioning to two next states, and determine the transition probability matrix A. Let the initial state be $\sigma_0 = (0, 0)$, and show that the steady-state distribution for states is equiprobable.

(b) Show further that all *pairs* of system outputs are equiprobable in the steady state.

2.5.10. In the channel with memory found in Example 2.23, find the joint probability of two consecutive errors, first by computing the steady-state probabilities of channel pair states and then using the specified probabilities of channel action conditioned upon state. Note that the probability of consecutive error is quite different than the square of the marginal error probability, indicating that the channel is not memoryless.

2.6.1. [Proof of (2.6.3) for binary case.] Define decision boundaries as in (2.6.3) and express the probability of error as the sum of integrals over two regions. Now move a portion Δ of one decision region into the other, as indicated in Figure P2.6.1. Express the resulting probability of error as a sum of the previous two integrals, less an integral over the perturbation region. Show by the likelihood ratio that this last integral is nonnegative; hence the new error probability can be no smaller than before.

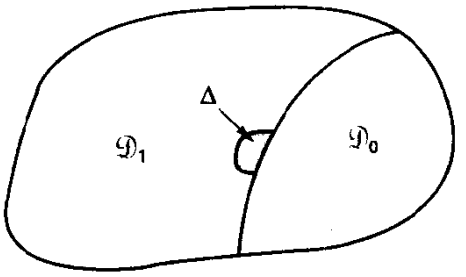


Figure P2.6.1

(b) Generalize this argument to show that the same decision rule is optimal in the M-ary case.

2.6.2. In Exercise 2.1.3, visualize the space of observations as the eight vertices of a cube and label each. Formulate the ML decision rule, invoking a tie-breaking procedure to break ties. Calculate the exact probability of error (by symmetry, each message will have the same result); then use a union bound to upper bound the probability of error. Compare the results.

2.6.3. We are given that $X \in \{1, -1\}$, that $Y = X + N$, where N is uniformly distributed on $[-2, 2)$, with N independent of X, and that the two possibilities for X are equiprobable. Show that the following decision rules all have the same error probability, $P(\epsilon) = 0.25$:

(a) Choose $X = -1$ if $Y \leq 0$; else choose $X = 1$.

(b) Choose $X = -1$ if $Y \leq -1$; else choose $X = 1$.

(c) Choose $X = -1$ if $Y \leq -1$; choose $X = 1$ if $Y > 1$; else decide with a toss of a fair coin.

2.6.4. Two tetrahedral-shaped dice, each labeled with 0, 1, 2, and 3, are in a box. (For such a die, we agree to observe the number on the bottom face!) One die is fair, and the other is loaded so that 0 is observed with probability $\frac{1}{2}$, the remaining numbers being equally likely. In N rolls of a die the probability of observing k_i occurrences of symbol i, where $k_0 + k_1 + k_2 + k_3 = N$, is given by the *multinomial distribution:*

$$P(k_0, k_1, \ldots, k_3) = \left[\frac{N!}{k_0! k_1! \cdots k_3!}\right] P_0^{k_0}, \ldots, P_3^{k_3}$$

where P_i are the probabilities of observing face i, conditioned upon which die was selected. The experiment consists of picking a die with equal probability, tossing it $N = 10$ times, and observing the number of appearances of each number. The task is to decide which die was chosen. Formulate the ML decision rule and show that it reduces to

$$\sum_{i=0}^{3} a_i k_i \underset{\text{loaded}}{\overset{\text{fair}}{\gtrless}} t,$$

where $a_i = \log_e(P_{i_{\text{fair}}}/P_{i_{\text{loaded}}})$. Find t. Given that 10 rolls produce observations of 5, 2, 1, and 2 of types 0, 1, 2, and 3, respectively, what would you conclude, and what is the a posteriori probability that in fact the loaded die was selected?

2.6.5. Compute the error probability for Example 2.26, assuming the same parameters for λ given there. In particular, calculate the probability that all seven counts of the noise-only slots are less than the count of the signal slot for a specific k_1; then weight these by the probability of obtaining $K_1 = k_1$ and add. Thus, we have

$$1 - P(\text{error}) = P(\text{correct}) = \sum_{k_1=0}^{\infty} P(k_1 \mid \text{signal}) \left[\sum_{k=0}^{k_1-1} P(k \mid \text{no signal})\right]^7.$$

(This is pessimistic with respect to ties.) Evaluate numerically.

2.6.6. A decision problem requires us to decide among two signal hypotheses. Under S_0, the observation is normal with zero mean and unity variance, while under S_1, the observation is normal with zero mean and variance 10. Ten independent observations are made. Assuming equiprobable selection of hypotheses, show the minimum probability of error rule reduces to

$$y = \sum_{n=1}^{N} r_n^2 \underset{S_0}{\overset{S_1}{\gtrless}} t,$$

where t is a threshold, so that the sums of squares of the observations forms a sufficient statistic. If $N = 10$, find t and the probability of error. *Hint:* The latter requires the fact that the decision statistic y have a chi-square distribution, with 10 degrees of freedom, and p.d.f. given by

$$f_{Y|\sigma}(y \mid \sigma) = \frac{y^4 e^{-y^2/2\sigma^2}}{2^5 4! \sigma^{10}}, \qquad y \geq 0.$$

2.6.7. For the signaling situation diagrammed in Figure P2.6.7, show that r_2 is *not* irrelevant to the decision process, although r_2 is comprised totally of noise. Assume both noise random variables are Gaussian with zero mean and unit variance and that the two noises are independent.

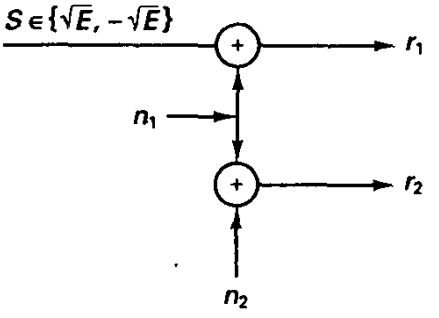


Figure P2.6.7

(a) Explain in heuristic terms what the use of r_2 provides. An intuitive way to process the data is to subtract r_2 from r_1, which provides a statistic with n_1 removed. Is this a sufficient statistic?

(b) Show that the optimal test statistic is of the form $T = r_1 + \alpha r_2$, which is to be compared with a threshold t.

(c) Express the probability of error in terms of the function $Q(x)$.

2.6.8. In the communication system depicted in Figure P2.6.8, intuition would suggest that r_2 is irrelevant, since this observation is merely a noisier version of r_1. Verify that this is indeed the case.

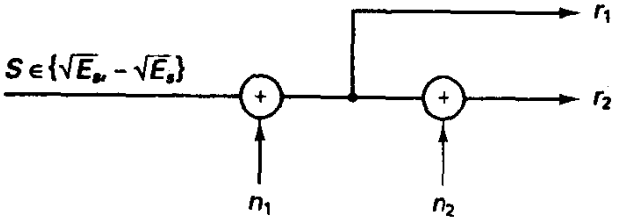


Figure P2.6.8

2.6.9. A binary hypothesis test is posed as follows: under the first hypothesis, the observation vector $\mathbf{r}$ is jointly Gaussian with mean vector $\mathbf{m}_0$ and covariance matrix $\mathbf{K}_0$. Under the second hypothesis, the observations are jointly Gaussian with mean vector $\mathbf{m}_1$ and covariance $\mathbf{K}_1$. Determine the form of the optimal decision, in particular showing that the test statistic T is a linear function of the observations and is compared with a threshold t. Interpret the partitioning of observation space by a hyperplane.

2.6.10. Consider the situation of Example 2.24, a Gaussian decision problem involving two signals. In this case we were able to exactly determine the error probability of the ML detector. As an alternative, compute the bound on error probability developed in (2.6.27) by substituting

for the two p.d.f.'s and simplifying by combining exponential forms. By completing the square and integrating, obtain an exponential bound involving the signal amplitude and the noise variance. Compare numerically with the exact result. You could verify that the proposed bound has the same exponential dependence as the exponential approximation to the Q-function result for the exact probability of error.

2.6.11. (After Wozencraft and Jacobs [5].) Suppose two messages are signified by the vectors $(-1, -1)$ and $(1, 1)$. The messages are equiprobable. To each coordinate is added independent double-exponential (or Laplacian) noise, with p.d.f. given by

$$f_N(n) = \frac{1}{2}e^{-|n|}.$$

Formulate the two conditional p.d.f.'s for the observation $\mathbf{r} = (r_1, r_2)$, and show that the decision regions are as shown in Figure P2.6.11. An equivalent test, if the messages are equiprobable, is

$$r_1 + r_2 \underset{<}{\overset{>}{}} 0$$

or decide in favor of the nearest (in the Euclidean sense) signal.

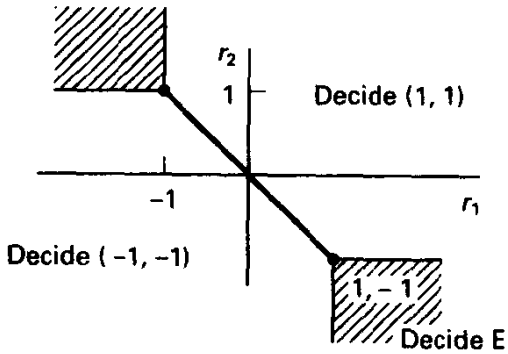


Figure P2.6.11

2.7.1. Prove (2.7.7) using the information theory inequality and the definition in (2.7.6).

2.7.2. Show that the maximum likelihood decoding rule has an alternative interpretation in terms of mutual information: Given reception of a specific $\mathbf{r}$, pick the $\mathbf{x}$ that maximizes $I(\mathbf{r}; \mathbf{x})$, the "event information."

2.7.3. A binary symmetric erasure channel (BSEC) is diagrammed in Figure P2.7.3. Determine the channel capacity C in terms of δ and ϵ and verify that the result reduces to the results stated in Section 2.7 for the BEC and BSC.

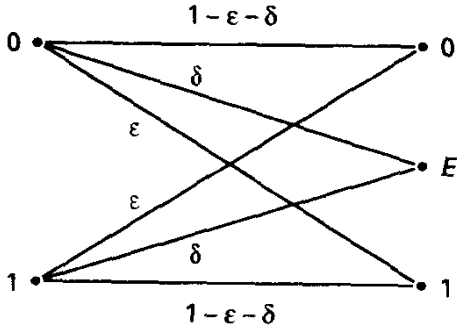


Figure P2.7.3

2.7.4. Consider the Z-channel with crossover parameter δ, having inputs 0 and 1 having probabilities q_0 and $1 - q_0$, respectively (see Figure P2.7.4). Find the capacity C by expanding the definition of mutual information and then maximizing with respect to q_0. Evaluate for $\delta = 0.1$. The optimal input distribution is more uniform than we might expect; it is tempting to say a good signaling strategy for this channel would send 0 often because it is unambiguously received. However, such a choice reduces the source entropy. You should also observe that with the optimal input probability assignment $I(x = 0; Y) = I(x = 1; Y) = C$; that is, each specific input selection has the same mutual information with the output ensemble Y. This is a necessary and sufficient condition (for each input that has nonzero probability) for attainment of capacity; see Theorem 4.5.1 of Gallager [12].

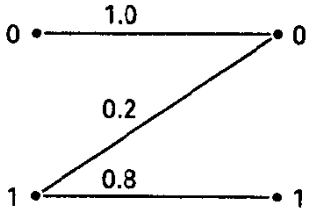


Figure P2.7.4

2.7.5. A binary input, eight-level output channel is depicted in Figure 2.7.5d. Show that the channel is symmetric, and determine the channel capacity C. Repeat if consecutive pairs of output symbols are merged into new symbols with probabilities obtained by summing the merged symbol probabilities, producing a 4-ary output channel. The capacity should be less, illustrating the data-processing lemma.

2.7.6. We have available a BSC with $\epsilon = 0.05$ that can be used at most two times per source symbol. We wish to communicate the output of a 4-ary, equiprobable, memoryless source. Apply the converse to the coding theorem to calculate a lower bound on symbol error probability that cannot be beaten by any source/channel coding scheme. Compare this result with the simple approach of assigning 2-bit tags to source symbols and transmitting these identifiers through the binary channel with no other coding.

2.7.7. Consider the three-input, three-output channel of Figure P2.7.7. This channel might seem symmetric by normal notions. Is it? Determine its channel capacity by guessing an input distribution and testing whether each input character supplies equal information with the output variable Y.

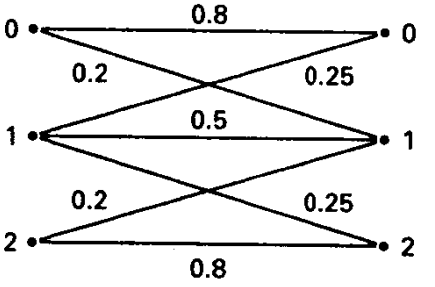


Figure P2.7.7

2.7.8. The five-input, five-output channel shown in Figure P2.7.8 is symmetric. Determine its channel capacity. We know this places an upper limit on the rate of transmission that achieves arbitrarily good reliability. Show that a simple block code that sends information in two-symbol blocks can achieve *zero* error probability while sending at a rate of $\log_2 5/2$ bits/code symbol.

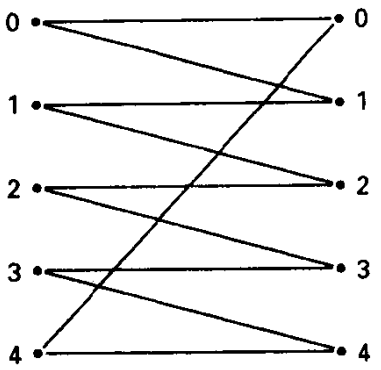


Figure P2.7.8

2.8.1. Consider a binary message source that produces 0 and 1 independently with probability 0.2 and 0.8, respectively.

(a) In messages of length 10, which of (0000000000), (0111110111), and (1111111111) would you consider "typical"?

(b) Suppose we provide codewords for all source sequences with 0, 1, 2, or 3 zeros in 10-symbol blocks, plus a codeword that represents any other outcome. How many codewords do we need, and what is the rate of this source code? What is the probability that a nonunique encoding occurs?

(c) By finding the source entropy, compute the approximate size of the typical message set for strings of length L, $2^{LH(U)}$, and express this as a fraction of the total number of possible messages, 2^L. (The notion of typicality is rather subtle, for although the all 1's sequence may be atypical by our notion of typicality it is more probable than the specific sequence we casually regard as typical!)

2.8.2. Apply the Kraft inequality to test whether a binary ($D = 2$) variable-length prefix code for $K = 8$ codewords is possible with lengths 1, 2, 3, 4, 5, 6, 7, and seven symbols. What about lengths 2, 2, 2, 3, 4, 5, 5, and 6? Draw code trees for each.

2.8.3. Zipf's law[25] states that words in a language, when ordered in decreasing relative frequency of usage, have probability law approximated by

$$P(\text{word } n) \sim \frac{K}{n}, \qquad n = 1, 2, 3, \ldots,$$

where n is the rank order and K is a constant. If we adopt this model for a vocabulary with 12,366 words and set $K = 0.1$, then the probabilities formally sum to near 1. Show numerically that the entropy of the word sequence, assuming independence, is 9.72 bits/word. If the average word length in English is 4.5 letters/word, then the entropy per letter is 2.16 bits/letter. This is more than 1 bit/letter less than an empirical result of Shannon based on trigrams, indicating the importance of incorporating as much structure as possible into source modeling.

2.8.4. A distant civilization has a four-letter alphabet $\{\Delta, *, !, \phi\}$.

(a) A cursory study of the language reveals marginal letter probabilities (0.5, 0.3, 0.15, 0.05), respectively. Design a Huffman code for this clan, and compare its efficiency with that which sends 2 bits per source symbol.

(b) Closer study of the language indicates a first-order Markov source model is more appropriate, with state transition diagram shown in Figure P2.8.4. Solve for the steady-state letter probabilities and the source entropy $H(X)$.

[25] G. K. Zipf, *Human Behavior and the Principle of Least Effort*, Addison-Wesley, Reading, MA, 1949.

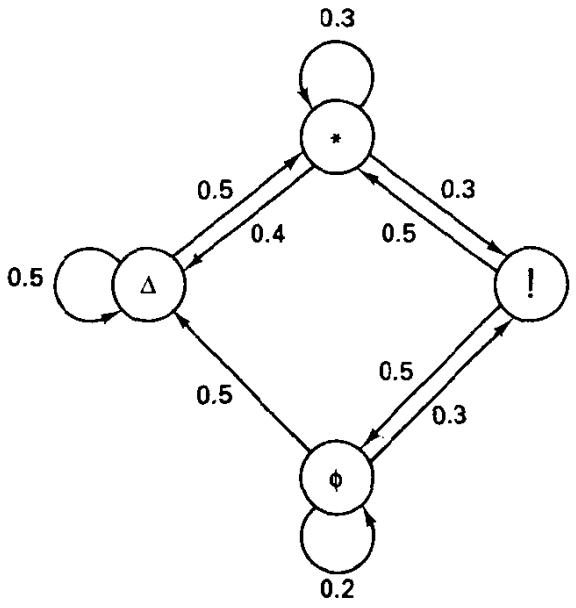


Figure P2.8.4

2.8.5. Design a Huffman code for the source described in Exercise 2.8.4 if we decide to encode $L = 2$ symbols at a time. You will need to identify the possible pairs of symbols and their probabilities prior to designing the code. Evaluate the efficiency of the code in average number of code bits per source symbol, and compare against the entropy limit for the Markov model. You should find that coding two symbols jointly is slightly more efficient than coding symbols individually with a Huffman code.

2.8.6. Design a $D = 4$-ary Huffman code for the discrete source of Example 2.31, and evaluate its performance relative to theoretical limits. (You should find $\overline{N} = 1.41$ code symbols/source symbol.)

2.8.7. (Following [12].) Run-length coding is a popular source coding technique for memoryless sources with highly skewed probabilities, or for sources with memory that exhibit long runs of identical symbols. We shall consider encoding of a binary memoryless source with $P(0) = 0.95$ and $P(1) = 0.05$. To encode, we begin counting consecutive occurrences of the most probable symbol. Let the number of counts until the next observance of 1 be designated $C, C = 1, 2, \ldots$. We pick some integer L, usually a power of 2. If the run count terminates at or before L, that is, $C \in \{1, 2, \ldots, L\}$, then we encode the count with 1 followed by the binary equivalent of $C - 1$, requiring $1 + \log_2 L$ code bits. A new count is then begun. If the run count reaches L, we send the code symbol 0 and begin the count anew.

(a) Convince yourself that the decoder can rebuild the original source string from the code string.

(b) To analyze performance, compute the probability that the run counts $C = 1, 2, \ldots, L$ occur, and thereby calculate the expected number of source symbols per run count. Likewise, calculate the expected number of code symbols per run count. For a long string of source outputs, appeal to the law of large numbers to argue that the average number of code symbols per source symbol is

$$\overline{N} = \frac{\text{expected number of code symbols/run count}}{\text{expected number of source symbols/run count}}$$

(c) Typically, the best choice for L is a power of 2 nearest $1/P$ (rare symbol), which in this case would suggest that $L = 16$. Evaluate the performance in this case.

2.8.8. The UNIX operating system utility "compress" uses a variation of the Lempel-Ziv-Welch algorithm for file compression. If such a system is available, experiment with some text files and determine file size before and after compression.

2.8.9. The lexicographer G. H. McNight observed in 1923 that in the English language 43 words, including "and, the, of, have, to, and you," constitute 50% of the words in standard text. Assume that the remainder of the dictionary of words is 8192 words, occurring equiprobably. View words as the source entities, and devise a simple source coding scheme that operates with 10.5 code bits per word.

2.9.1. For the additive Gaussian noise channel with input energy constraint, $C = \frac{1}{2}\log_e \left[1 + (E/\sigma^2)\right]$ nats, achieved for any E by an input selection that is Gaussian with variance E. Show that if the SNR per use of the channel is small, that is, $E \ll \sigma^2$, binary signaling with $\pm\sqrt{E}$ inputs, equiprobably chosen, essentially achieves capacity.

(a) Formulate mutual information for the binary input, Gaussian noise channel model. Note, by symmetry, that C is achieved with equiprobable inputs and hence may be evaluated from

$$C = \int_{-\infty}^{\infty} f(y \mid x = E^{1/2}) \log_e \left[\frac{f(y \mid x = E^{1/2})}{\frac{1}{2} f(y \mid x = E^{1/2}) + \frac{1}{2} f(y \mid x = -E^{1/2})} \right] dy \quad \text{nats}$$

Evaluate this numerically for $E/\sigma^2 = 0.1, 0.2, 0.5, 1.0, 2.0, 5.0,$ and 10.0, and compare with capacity without the binary input assumption. By expansions for the logarithm function, we may determine analytically that C approaches $E/2\sigma^2$ nats for small SNR in both cases.

2.9.2. Suppose the channel is as described in Exercise 2.9.1, but we place a binary quantizer (sign detector) on the channel output. This converts the previous channel into a BSC.

(a) Show that the channel error probability is $\epsilon = Q(E^{1/2}/\sigma)$.

(b) Use this ϵ in the expression for capacity of a BSC, and evaluate for E/σ^2 values given previously. You should find that capacity is somewhat less with such channel quantization.

2.9.3. In Example 2.34, we calculated the channel capacity for a Gaussian noise channel under an energy constraint on the input. Since the resulting capacity for that problem was under 2 bits/channel use, let's see what happens if we use a four-level input to the channel, with levels $\pm A, \pm 3A$.

(a) Let the p.m.f. for this discrete distribution be equiprobable and chosen to meet the energy constraint; that is, $\frac{2}{4}A^2 + \frac{2}{4}9A^2 = E$. Suppose the output Y is quantized with a uniform four-level quantizer having thresholds placed midway between the conditional means of Y. Calculate the resulting channel transition probabilities and then the mutual information achieved with this system.

(b) Can you think of a way to improve on the strategy still using four-level inputs?

2.9.4. A vector Gaussian channel is available for which the four component channels have noise variances of 1, 2, 4, and 8 units, respectively. Let the available energy allocation be 6 units. Find the optimal energy allocation and the resulting capacity in bits per vector channel use. Repeat for an energy allocation of 12 and 18 units.

2.9.5. Evaluate the channel capacity for the following waveform channels, assuming that the noise is additive white Gaussian noise.

(a) The dial-up telephone channel modeled as band-limited to [300, 3000] Hz, and the S/N measured over this bandwidth is 1000, or 30 dB.

(b) The deep-space communication channel with the *Voyager* spacecraft at Jupiter encounter: bandwidth is unconstrained, and $P/N_0 = 10^6$.

(c) A satellite communication channel operating with a transponder bandwidth of $B = 36$ MHz and with $P/N_0 = 5 \cdot 10^8$.

2.9.6. In the text we derived a lower bound on E_b/N_0 for reliable communication on the band-limited Gaussian noise channel:

$$\frac{E_b}{N_0} \geq \frac{B}{R_b}(2^{R_b/B} - 1)$$

for all systems having a bandwidth ratio of B/R_b.

(a) Plot this lower bound as a function of bandwidth ratio, and note that as this ratio increases without bound the minimum E_b/N_0 approaches -1.6 dB. Also, find the necessary bandwidth ratio for which the required SNR is only 1 dB larger than the infinite bandwidth limiting value.

(b) A newly hired engineer offers a design for sending "error-free" data at 1 Mbps through a 100-kHz bandwidth Gaussian channel, doing so with $E_b/N_0 = 25$ dB. Can the engineer be correct?