4

Channel Coding and Its Potential

Almost all codes are good, except those we can think of!

Attributed to the late J. Wolfowitz

In Chapter 3 the performance of many signaling formats has been analyzed for important channel models and different detection strategies. The point of view there was that each symbol produced by the modulator was a message in itself, independent of previous and future transmissions, observed in the presence of white Gaussian noise. Thus, in the absence of intersymbol interference effects, the detector can optimally decide each symbol by itself, as in one-shot transmission.

The process of ***channel coding*** produces modulator input symbols that are interrelated in either a block-by-block or sliding-window fashion, introducing a crucial aspect of memory into the signaling process. At the same time, there is introduced a controlled redundancy, in that the number of actually producible waveforms in a given interval is less than that which could be produced by the same modulator when no coding is employed.

The reasons for adopting coding are, broadly speaking, to achieve highly reliable communication at rates approaching the channel capacity limit defined by the physical channel and to do so in an instrumentable way. We have, for example, determined that orthogonal signal sets achieve the Shannon capacity limit for the infinite bandwidth

Gaussian noise channel as M becomes very large, but the demodulator complexity per bit grows essentially exponentially with M, as does the bandwidth, and we do not regard this as an attractive solution. The channel coding approach offers the same potential performance, in principle, through construction of elaborate signal sequences lying in high-dimensional spaces, but composed from elementary modulator sets. A now classical example is the use of binary channel encoding functions, with code symbols communicated using antipodal signaling, where signal sequences can be viewed as occupying a (sparse) set of the vertices of a high-dimensional cube. The net result is that spectral occupancy and demodulator/decoder complexity can be far less than the orthogonal construction would imply for equivalent levels of performance.

Channel coding is useful in virtually every kind of noisy channel transmission problem; some still regard its principal area of application as the unlimited-bandwidth channel, but recently major contributions to practical communications have been made by intelligent coding for band-limited channels. We will also find coding offers particularly impressive gains on fading and time-varying interference channels.

Our first section in this chapter is a description of generic channel coding approaches to provide the reader with general familiarity and a preview of material to follow in Chapters 5 and 6. Beyond these fundamental notions, however, we are not presently interested in the exact construction of codes. Instead, the major theme of the chapter deals with the information-theoretic potential of coding, without resort to description of best codes. With the converse to the coding theorem presented in Chapter 2, we have demonstrated that it is impossible to transfer information faster than the channel capacity limit, C, measured in bits per channel use, with vanishingly small error probability. The positive side of the argument, that if the information rate R is strictly less than C arbitrarily reliable communication is achievable, is referred to as the direct noisy channel coding theorem and is the subject of a major part of this chapter. In developing this result, we shall also encounter the parameter R_0, which serves as a compact figure of merit for a modulation and demodulation system when coding is employed. The importance of R_0 to coded systems was first advanced by Wozencraft and Kennedy [1] and later by Massey [2]. The remainder of the chapter examines in detail the R_0 viewpoint toward communications as a modern means of assessing different modulation and coding options.

4.1 A TAXONOMY OF CODES

At the heart of any coding technique is a mapping from sequences of message symbols to sequences of input labels to the modulator, which in turn produces a sequence of modulator signals uniquely determined by the input to the encoder. Coding techniques may be classified based on the structure behind the encoding function, that is, the relation between message symbols and modulator inputs. The first distinction, shown in Figure 4.1.1, is between ***block codes*** and ***trellis***, or ***sliding block codes.*** Both may be viewed as mappings from the space of discrete-alphabet input sequences, called messages, to the space of discrete-alphabet output sequences, called codewords or code sequences. Frequently, but not always, the two alphabets are the same.

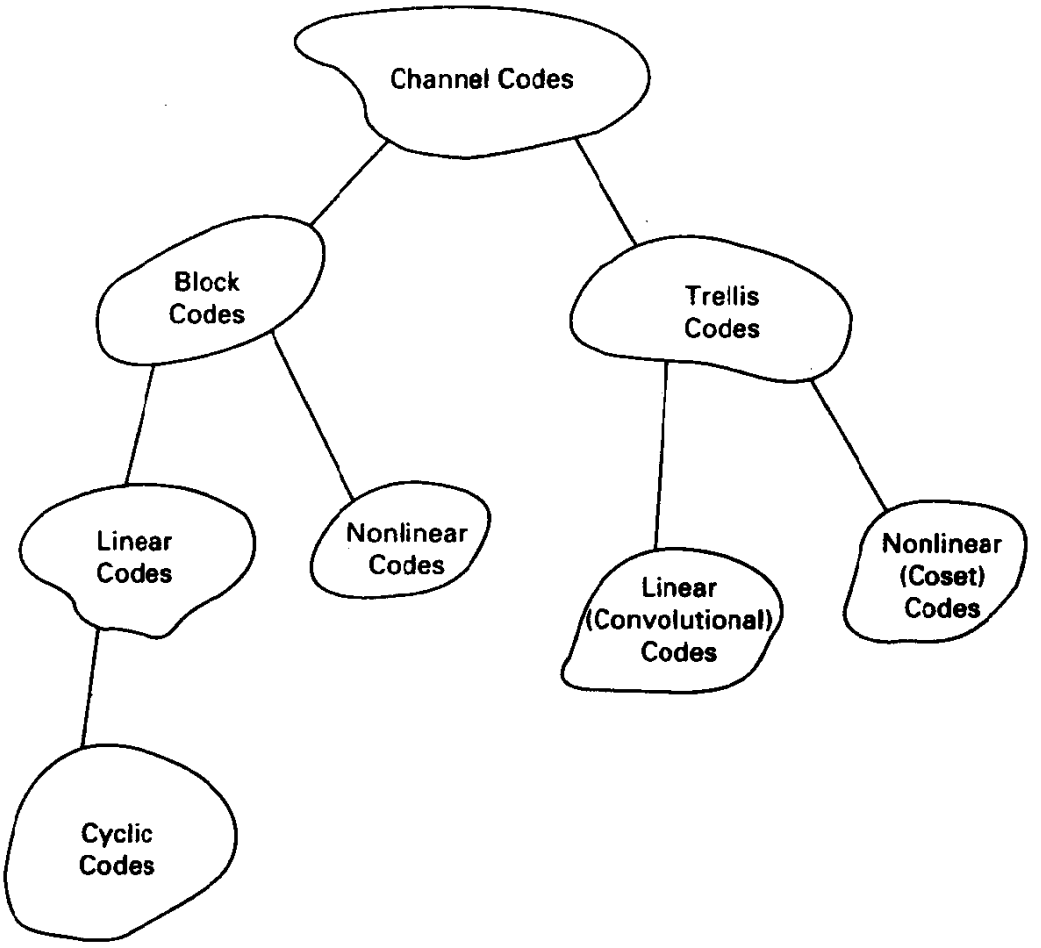

Figure 4.1.1 Taxonomy of channel coding techniques.

As the name connotes, ***block codes*** operate in block-by-block fashion, and each codeword depends only on the current input message block. We may view the code as a dictionary of codewords addressed by input messages. Block codes may be further categorized as ***linear*** or ***nonlinear*** codes. Linear codes are defined by a linear mapping (over an appropriate algebraic system) from the space of input messages to the space of output messages, and this is ultimately represented by a matrix multiplication. As we shall see, this algebraic structure allows significant simplification of encoding and decoding equipment. Linear codes are also known as parity check codes because we can view the codeword as comprised of a message component and parity symbols, analogous to a single parity bit used in simple error-checking systems. Nonlinear codes, although not particularly important in the context of block coding, are the remaining codes. The practically important linear codes are in a more restricted class known as ***cyclic codes,*** or at least codes closely related to cyclic codes. Their cyclic structure admits still further hardware or software simplifications. These properties will be further developed in Chapter 5.

Trellis encoders, in contrast, should be viewed as mapping an arbitrarily long input message sequence to an arbitrarily long code stream without block structure. The output

code symbol(s) at a certain time is defined to depend on the ***state*** of a finite-state encoder, as well as on current inputs. Since the encoder state is normally specified by a short block of previous inputs, the name sliding-block code is sometimes used. In practice, messages and code sequences are terminated at some point, in which case we could say we have produced a (long) block code. However, the description and instrumentation of trellis codes are quite unrelated to this observation. Trellis codes get their name because the codewords may be identified with a regular, directed finite-state graph reminiscent of a garden trellis, a concept introduced by Forney [3].

Linear trellis codes are known as ***convolutional codes,*** because the code sequence can be viewed as the convolution (in discrete time and over a discrete alphabet) of the message sequence with the encoder's impulse response. In practice, most trellis codes have thus far been linear codes, but this linear/nonlinear option does *not* have significant impact for maximum likelihood decoding in the case of trellis codes. The complexity of the ML decoder depends only on the number of states in the encoder, whether or not the encoder implements a linear mapping. It is true that the design and analysis of codes is simplified by the linearity property, and some simple decoding procedures (for example syndrome decoders) require the linear structure. This will be revisited in Chapter 6.

So, what is the underlying thought behind coding? Why bother with the complexity? Both kinds of codes install two key features into the code sequence: ***redundancy*** and ***memory.*** Redundancy means that the set of allowable code sequences, or codewords, is smaller (often many orders of magnitude smaller) than the number of sequences suggested by the size of the code alphabet. Thus the code symbols do not carry as much information per symbol as they might without coding, and we speak of the transmissions as being redundant. This redundancy may accomplish little, however, unless the code symbols depend on many input symbols, which we could ascribe as memory. Equivalently, the information sequence is somehow diffused throughout the code sequence. The combination of the two features allows the decoder to use sequence observations to make more reliable decisions about the original message by exploiting the averaging tendency associated with the law of large numbers. This will become evident shortly.

Prior to Shannon's work, communications engineers understood a fairly obvious fact—that redundancy was useful at increasing reliability, in the form of repeating the message several times, hoping to get it correct by majority voting among successive decisions. The problem is that this repetition reduces the information throughput per channel use. The missing conceptual ingredient was that encoding and decoding with memory could avoid this large penalty in throughput while still maintaining high reliability. Shannon showed that, as long as the message has a sufficiently small attempted throughput per channel use, then high reliability is possible. How small is small enough? Channel capacity is the magic number!

In the roughly 45 years since Shannon's paper appeared, both block codes and trellis codes have had their share of advocates, and the debate over the relative merits of the two classes of codes has been occasionally heated and usually entertaining. One wag has joked that "block codes make for good papers, but trellis codes make for better sales." Both types of codes have their own advantages in certain applications, which will become clear in the next two chapters, and it is essential for the communication engineer to be fluent in the language and principles of both. Actually, there is more congruence between block and trellis codes than commonly realized, which we shall try

to illuminate, and, for that matter, some of the most powerful approaches in use today utilize block and trellis codes in a concatenated, or hierarchical, manner.

In the following sections, we shall further develop the coding potential of block codes for no other reason than that the block structure is simpler to visualize and analyze. Our objective is not to highlight specific techniques, which we shall study in Chapter 5, but to glimpse the real promise of information theory for reliable digital transmission.

4.2 INTRODUCTION TO BLOCK CODING AND OPTIMAL DECODING

A block code C is merely a list of T codewords, $\mathbf{x}_i, i = 1, \ldots, T$, each an n-tuple whose entries are from an alphabet of size q. These codewords are to be used for representing one of T messages, and assuming that the message source selects messages equiprobably and independently from message to message, the entropy of the codeword selection process is $\log_2 T$ bits per message.

The codewords are injected into the available channel by some digital modulation process (often the alphabet size q matches that of the chosen modulator), and we assume for now that the cascade of modulator/channel/demodulator is a memoryless channel, perhaps a discrete-output channel. The information exchanged between source and user, if no uncertainty remains after observing the channel output sequence $\mathbf{y}$, is $\log_2 T$ bits, or $(\log_2 T)/n$ bits per codeword symbol. We define the latter as the ***rate*** of the code:

$$R = \frac{\log_2 T}{n} \quad \text{information bits per channel symbol.} \tag{4.2.1}$$

Alternatively, a code of rate R and block length n has $T = 2^{nR}$ codewords.

For example, if we generate a table having $T = 1024$ codewords of binary 15-tuples, this forms a code of rate $R = \frac{10}{15} = \frac{2}{3}$ bits/channel symbol. A rate $R = \frac{1}{2}$ code with codewords each $n = 80$ bits long would have 2^{40} codewords! This, in fact, is not at all a large code by modern standards, which suggests something other than table-lookup encoding and decoding must be employed. It is interesting to note that, even for this relatively modest coding arrangement, communication of the entire set of codewords at a source rate of 1 Gbps would require duration of many orders of magnitude longer than the age of the universe!

We should emphasize that no special mathematical structure has been imposed on the code at this point, although we will do so in Chapter 5 when dealing with actual implementations. For the present, a code is simply a dictionary, or lexicon, relating messages to codewords.

Now consider the situation shown in Figure 4.2.1. The message source selects a message, say the ith message, to which is associated a codeword, $\mathbf{x}_m$. We will concentrate initially on the case where each code symbol of the selected codeword is acted on by a discrete memoryless channel (DMC) with a q-ary input alphabet and Q-ary output alphabet, where $Q \geq q$. The physical origins of this channel are not important for the present. The channel is completely specified by input/output transition probabilities $P(y|x)$. The decoder seeks a minimum-probability-of-error decision, based on the sequence $\mathbf{y}$, about which codeword was transmitted and, thereby, which message was

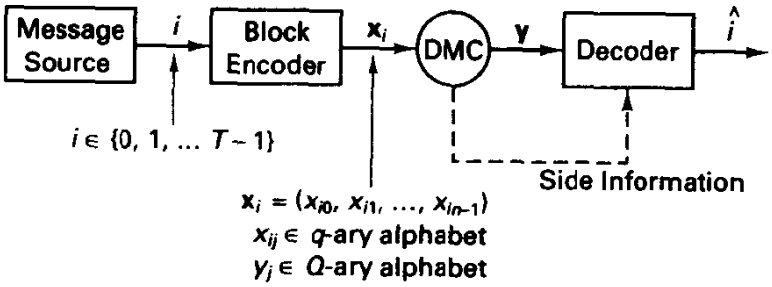

Figure 4.2.1 Block coding framework for DMC.

sent. Assuming that codewords are selected with equal probability, the best rule is, as in Chapter 2, the ML rule; that is, find that $\mathbf{x}_m$ which maximizes $P(\mathbf{y}|\mathbf{x})$. Because of the memoryless channel assumption, we may express this as

$$\max_{\mathbf{x}_i \in C} P(\mathbf{y}|\mathbf{x}_i) = \max_{\mathbf{x}_i \in C} \prod_{j=0}^{n-1} P(y_j|x_{ij}), \tag{4.2.2a}$$

where x_{ij} is the jth symbol in the ith codeword. Extensions of this basic model include the case of vector-valued continuous r.v. outputs from the channel, collected as $\tilde{\mathbf{y}} = (\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{n-1})$. We would then express the task as

$$\max_{\mathbf{x}_i \in C} f(\tilde{\mathbf{y}}|\mathbf{x}_i) = \max_{\mathbf{x}_i \in C} \prod_{j=0}^{n-1} f(\mathbf{y}_j|x_{ij}). \tag{4.2.2b}$$

Still another extension is the situation where the channel, or more precisely the demodulation equipment, supplies "side information" about the channel state(s) during the duration of a codeword. This side information is employed to construct the relevant likelihood functions for a given time index. We will return to such cases at the end of the chapter.

Returning to the DMC case, we can just as well take logarithms (to any base) of the product in (4.2.2), obtaining the equivalent rule

$$\max_{\mathbf{x}_i} \sum_{j=0}^{n-1} \log P(y_j|x_{ij}) = \max_{\mathbf{x}_i} \sum_{j=0}^{n-1} \lambda(y_j, x_{ij}) = \max_{\mathbf{x}_i} \Lambda(\mathbf{y}, \mathbf{x}_i). \tag{4.2.3}$$

Here we have introduced the notion of a symbol ***metric,*** $\lambda(y_j, x_{ij}) = \log P(y_j|x_{ij})$, which scores each code symbol by the ***log likelihood.*** The total score, or metric, for a codeword, $\Lambda(\mathbf{y}, \mathbf{x}_i)$, is the sum of these metrics.

Occasionally, for reasons of convenience, metrics other than the optimal log-likelihood metric are used. However, for many channels and modulation formats, the optimal metric is easy to determine and implement, as the next two examples illustrate.

Example 4.1 Maximum Likelihood Decoding on the BSC

Consider transmission of binary code symbols through a BSC, which might arise from a variety of binary modulation and detection options. Let ϵ be the channel error probability.

The likelihood function for sequences, $P(\mathbf{y}|\mathbf{x}_i)$, is given by

$$\begin{aligned} P(\mathbf{y}|\mathbf{x}_i) &= \prod_{j=0}^{n-1} P(y_j|x_{ij}) \\ &= \epsilon^{d_H(\mathbf{y},\mathbf{x}_i)}(1-\epsilon)^{n-d_H(\mathbf{y},\mathbf{x}_i)}, \end{aligned} \tag{4.2.4}$$

where the function $d_H(\mathbf{y}, \mathbf{x}_i)$ merely counts the number of places where the vectors $\mathbf{y}$ and $\mathbf{x}_i$ differ. This function is known as the ***Hamming distance*** between vectors and is fundamental to the study of coded systems. Because of its central importance, we repeat: ***the Hamming distance between two n-tuples over the same alphabet (binary or otherwise) is the number of positions where the vectors are not equal.*** Hamming distance is a true distance measure in the mathematical sense, satisfying requirements of nonnegativity, symmetry, and the triangle inequality. (See Exercise 4.3.1.)

Returning to the decoding task, it is obvious by taking the logarithm of (4.2.4) that the log-likelihood metric is

$$\Lambda(\mathbf{y}, \mathbf{x}_i) = \log P(\mathbf{y}|\mathbf{x}_i) = d_H(\mathbf{y}, \mathbf{x}_i) \log \frac{\epsilon}{1-\epsilon} + n \log(1-\epsilon). \tag{4.2.5}$$

The second term in (4.2.5) may be discarded since it contributes equally to all codeword metrics. Thus, maximum likelihood decoding on the BSC corresponds to *minimum* Hamming distance decoding, provided $\epsilon \leq 1/2$. We can add any constant to the log likelihood, as well as scale by any positive constant, without affecting the outcome. Hence, we could assign the per-symbol metric

$$\lambda(y_j, x_{ij}) = \begin{cases} 0, & y_j = x_{ij}, \\ -1, & y_j \neq x_{ij} \end{cases} \tag{4.2.6}$$

and find the code vector $\mathbf{x}_i$ whose metric $\Lambda(\mathbf{y}, \mathbf{x}_i)$ is largest. More typically, we adopt a symbol metric

$$\lambda(y_j, x_{ij}) = \begin{cases} 0, & y_j = x_{ij}, \\ 1, & y_j \neq x_{ij} \end{cases} \tag{4.2.7}$$

and choose that codeword with *smallest* metric sum, which translates to performing minimum Hamming distance decoding. This principle generalizes to any q-ary uniform channel, but not to q-ary symmetric channels, as discussed in the next example.

Example 4.2 Decoding of 8-PSK with Hard-decision Demodulation

Suppose that codewords are formed from an 8-ary alphabet and are communicated using 8-PSK modulation. Let the channel model be AWGN, and suppose the demodulator forms a hard decision on each code symbol, forming the best estimate of each symbol by itself. The resulting channel is the 8-ary symmetric channel shown in Figure 4.2.2. Notice that all transition probabilities are not equal in the error set, since, for example, adjacent symbol errors are more likely in M-PSK transmission than, say, antipodal error types.[1] Nonetheless, we can score each received symbol against a hypothesized code symbol with the metric $\log P(y_n|x_n)$, where both y_n and x_n are in the set $\{0, 1, \ldots, 7\}$. Here it is not simple Hamming distance between observed and test symbols that provides the optimal metric. If desired, the metrics, which are real numbers, may be scaled, translated, and rounded to integers without substantial loss in performance.

[1] This constitutes an example of a symmetric, but nonuniform channel.

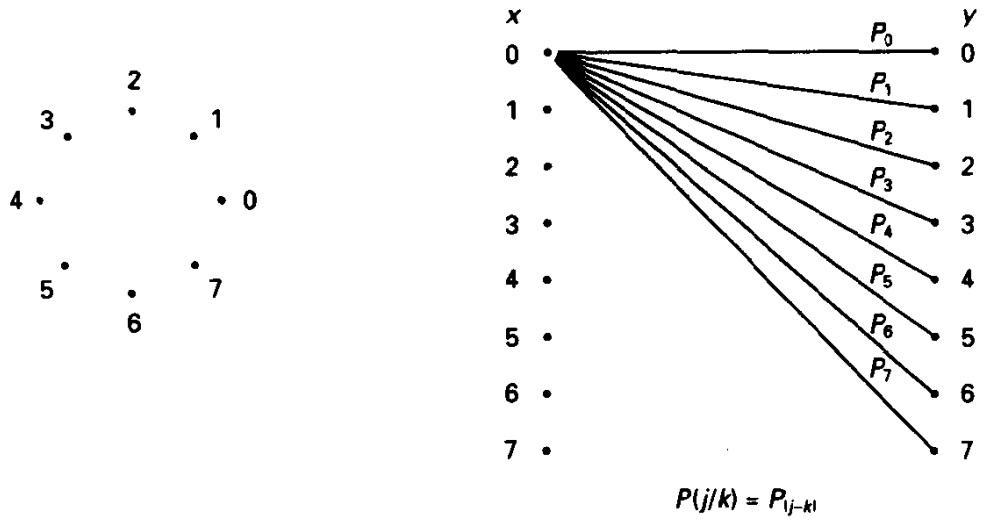

Figure 4.2.2 Eight-input, 8-output symmetric channel for hard-decision 8-PSK transmission.

Example 4.3 Soft-decision Decoding on a Gaussian Channel

Suppose that antipodal signaling is employed for code symbols and that the symbol energy-to-noise density ratio is a rather low $E_s/N_0 = 1/2$, or -3 dB. Rather than performing a hard decision on each code symbol, which would have error probability $Q(1) = 0.1587$, let the demodulator quantize the correlator output to eight levels, with level spacing $0.5\ E_s^{1/2}$. This constitutes a symmetric 2-input, 8-output discrete channel, with transition probabilities 0.308, 0.192, 0.192, 0.150, 0.0916, 0.0442, 0.0171, and 0.00598. (These are obtained using integrals of Gaussian p.d.f.'s.)

A decoder, when testing a certain binary symbol, should employ the log-likelihood metric

$$\lambda(y_j, x_{ij}) = \log P(y_j|x_{ij}), \tag{4.2.8}$$

which would be among the values -1.18, -1.65, -1.65, -1.90, -2.39, -3.12, -4.07, and -5.12 when natural logarithms are employed. In practice, these metrics would be translated and scaled so that low-precision arithmetic is possible.

Decoding with finely quantized demodulator outputs is known as ***soft-decision decoding*** in the literature, and at least on the Gaussian channel it buys important improvements in the energy efficiency as we will see. The maximum-likelihood rule forms a partition of observation space, and soft-decision decoding forms decision boundaries that are closer to the boundaries for the unquantized channel than with hard-decision decoding. To illustrate, suppose that the two codewords are $(0, 0, 0)$ and $(1, 1, 1)$. Example 2.24 showed that the optimal decision boundary is a plane bisecting the line connecting the two signals in signal space. Soft-decision decoding classifies a vector (y_0, y_1, y_2) according to the sum of log likelihoods, as before, with the resulting decision boundary shown in Figure 4.2.3. Near the origin, where the connecting line bisects the plane, the corrugated surface of Figure 4.2.3 is clearly a reasonable approximation to the ideal and much closer than the hard-decision boundary shown in Figure 2.6.6. Of course, if we had chosen the quantizer boundaries differently, the surface changes, pointing to the need to perform quantization carefully.

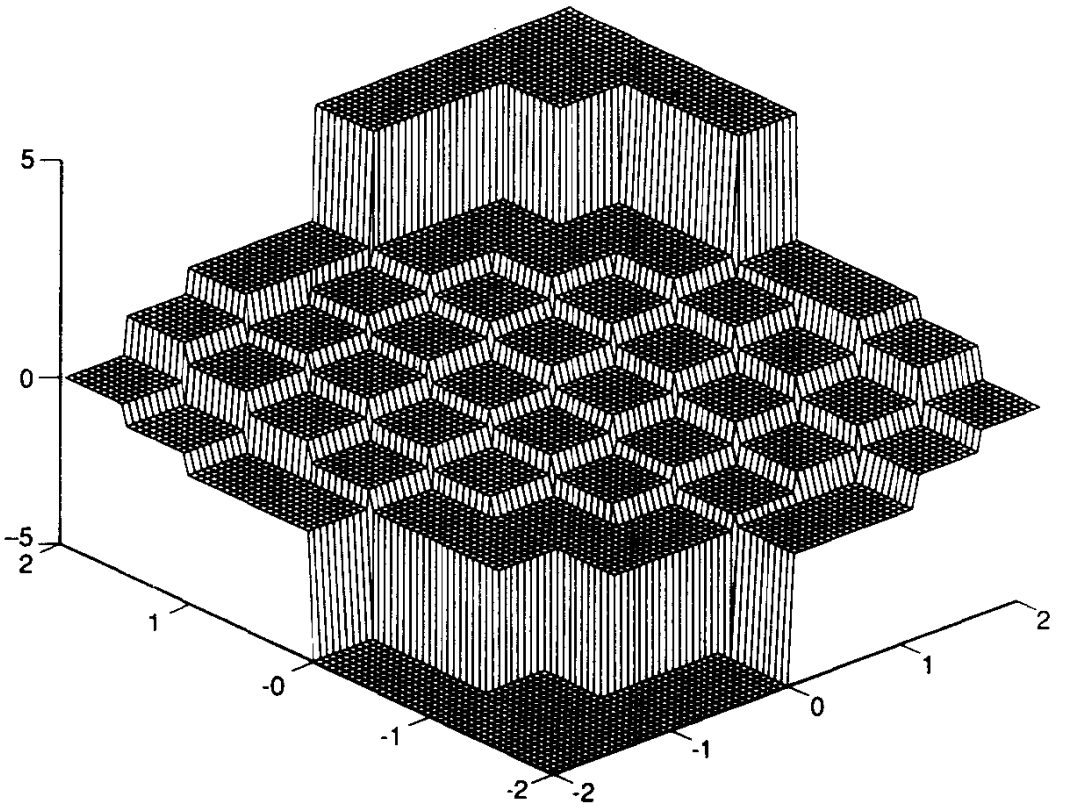

Figure 4.2.3 Decision surface for two codewords, $\mathbf{x}_1 = (-1, -1, -1)$, $\mathbf{x}_2 = (1, 1, 1)$, 8-level quantization.

Example 4.4 Decoding on a Fading, Noise-varying Channel with Antipodal Signaling

To consider a more complicated example, one with continuous variable channel outputs, suppose the codewords are binary; that is, $x_n \in \{0, 1\}$ and that code symbols are transmitted with an antipodal modulation scheme, say PSK, and that coherent demodulation is performed. Furthermore, we let the channel act on each symbol with a gain factor a_j and assume that the demodulator output carries a time-dependent Gaussian noise with variance σ_j^2, as depicted in Figure 4.2.4. We suppose the noise is independent from symbol to symbol. This situation might arise physically from a fading channel, with time-varying noise level due to pulsed jamming. We assume that both channel gain and noise level are *known* by the decoder, an example of the side information mentioned earlier.

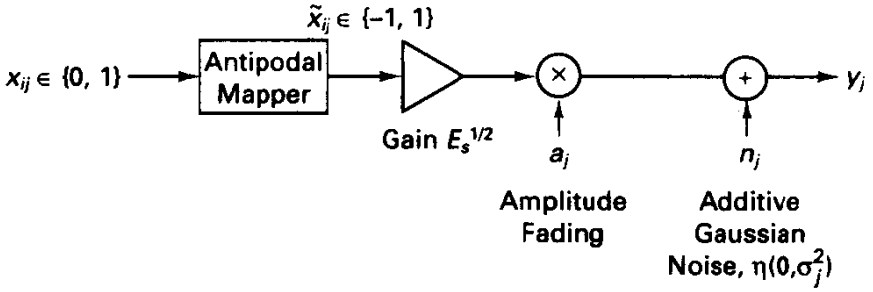

Figure 4.2.4 Channel model for Example 4.3.

The observation y_j is a Gaussian random variable whose mean is given by $\pm a_j E_s^{1/2}$, depending on the code symbol sent, and whose variance is σ_j^2. The likelihood function then is, due to independence of successive transmissions,

$$f(\mathbf{y}|\mathbf{x}_i, \{a_j\}, \{\sigma_j^2\}) = \prod_{j=0}^{n-1} \frac{1}{(2\pi\sigma_j^2)^{1/2}} e^{-(y_j - a_j \tilde{x}_{ij} E_s^{1/2})^2/2\sigma_j^2}, \tag{4.2.9}$$

where we have mapped x_{ij} to $\tilde{x}_{ij}$ by

$$\tilde{x}_{ij} = 2x_{ij} - 1. \tag{4.2.10}$$

(This relation takes $\{0, 1\}$ code symbols into $\{-1, 1\}$ modulator inputs.)

After forming the logarithm of (4.2.9) and eliminating terms that either do not involve the codeword index i or are the same for both modulator symbols, we find that the per-symbol metric should be

$$\lambda(y_j, x_{ij}) = \frac{a_j y_j \tilde{x}_{ij}}{\sigma_j^2}. \tag{4.2.11}$$

Thus, the optimal codeword metric is a weighted correlation of the real codeword sequence and the real-valued output of the channel, with weighting proportional to signal amplitude and inversely proportional to noise variance. The nonfading, fixed-noise-level Gaussian channel is obviously a special case of the model here, and in that case all weighting factors can be removed. Then signed addition of the demodulator outputs is the maximum likelihood decoding procedure.

Let's now return to the general coding/decoding task. Looking beyond the decoding complexity of performing the maximization in (4.2.3), we inquire about the probability of a decoding error. The ML decision rules given previously imply a partition of observation space into decision zones $D_i, i = 1, 2, \ldots, T$. Letting $\hat{\mathbf{x}}$ denote the decoder's choice of codeword, we write the error probability as

$$P(\hat{\mathbf{x}} \neq \mathbf{x}) = \sum_{i=1}^{T} P(\mathbf{x}_i) P(\hat{\mathbf{x}} \neq \mathbf{x}_i | \mathbf{x}_i \text{ sent}) = \sum_{i=1}^{T} P(\mathbf{x}_i) P(\mathbf{y} \in D_i^c | \mathbf{x}_i \text{ sent}), \tag{4.2.12}$$

where D_i^c is the complement of the decision region for codeword $\mathbf{x}_i$. In general, the conditional error probabilities in (4.2.12) may vary among codewords. Unfortunately, it is quickly apparent that (4.2.12) is difficult to evaluate exactly, even for simple situations with highly symmetric codes used on simple channels such as the BSC. Moreover, the code design problem is to specify the code that minimizes the probability of error (4.2.12). This is generally even more formidable. Shannon cleverly avoided this difficulty by not tackling head on the exact analysis of a given code and design of the "best" code, but instead analyzing the ensemble of all codes with a given set of code parameters (rate, alphabet, and block length). He was able to prove certain behavior for this *ensemble*: if $R < C$, the channel capacity defined earlier, then the probability of error, averaged over the ensemble of codes, diminishes to zero as n increases. Since at least one code in the ensemble of codes must be as good as the average, this clever argument proves the existence of good codes without ever finding them. Researchers later showed that this convergence of error probability to zero happens exponentially fast with block length, but the fundamental breakthrough was to show that if the attempted rate is less than

capacity then arbitrarily reliable communication is possible. Conversely, if rate exceeds capacity, we have already seen in Chapter 2 that the performance cannot be arbitrarily good.

The central problem of coding theory since 1948 has been to find easily implementable codes that approach the kind of performance that Shannon's early work promised. This *constructive coding* is the topic of Chapters 5 and 6, where we discuss the specifics of block and trellis codes. For the remainder of this chapter, however, we further investigate this behavior of code ensembles and introduce another powerful descriptor of a modulation-channel-demodulation system, called R_0, which is now widely employed in communication system analysis. We will study the implications for intelligent design of coded communication systems, based on channel capacity and R_0 considerations.

4.3 TWO-CODEWORD ERROR PROBABILITY AND R_0

To approach the problem of bounding the error probability for general codes, we first consider *two specific codewords* $\mathbf{x}_1$ and $\mathbf{x}_2$, with block length n, used on a memoryless channel. The channel output can either be discrete or continuous, although we will emphasize the discrete case at the outset.

We compute the probability of the event that $\mathbf{x}_1$ is transmitted, but $\mathbf{x}_2$ has higher likelihood, when computed by (4.2.2) or (4.2.3), as a result of channel imperfections. We write this error probability as

$$P_2(\mathbf{x}_1 \to \mathbf{x}_2) = \sum_{\mathbf{y} \in D_2} P(\mathbf{y}|\mathbf{x}_1), \tag{4.3.1}$$

where we interpret the summation as an n-dimensional sum and denote the decision region for codeword $\mathbf{x}_2$ by D_2. (See Figure 4.3.1.) In (4.3.1) we are simply totaling the probability of having received any $\mathbf{y}$ in the error set, given transmission of $\mathbf{x}_1$.

We have earlier indicated that the exact evaluation of error probabilities is a tough task in general, and we shall settle for an upper bound to (4.3.1). To do so, we can

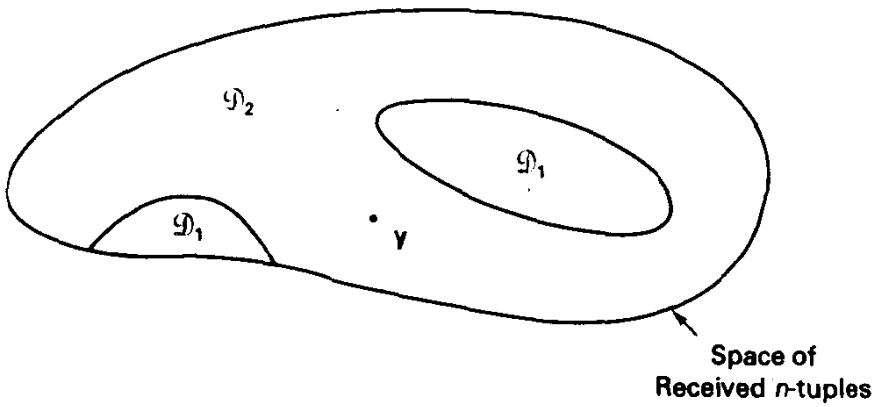

Figure 4.3.1 Decision regions for two-codeword problem.

multiply every term in (4.3.1) by a number larger than or equal to 1. For all $\mathbf{y} \in D_2$, $P(\mathbf{y}|\mathbf{x}_2) \geq P(\mathbf{y}|\mathbf{x}_1)$, by definition of the error region. Thus, we choose (with some hindsight) to multiply each term in the sum by

$$g(\mathbf{y}) = \left[\frac{P(\mathbf{y}|\mathbf{x}_2)}{P(\mathbf{y}|\mathbf{x}_1)}\right]^{1/2} \tag{4.3.2}$$

which also is greater than or equal to 1 for all $\mathbf{y}$ in the range of the sum. Doing so, we find that

$$P_2(\mathbf{x}_1 \to \mathbf{x}_2) \leq \sum_{\mathbf{y} \in D_2} P(\mathbf{y}|\mathbf{x}_1)^{1/2} P(\mathbf{y}|\mathbf{x}_2)^{1/2} \tag{4.3.3}$$

We can retain an upper bound by including in the summation *all* $\mathbf{y}$'s, not just those in the region D_2, yielding

$$P_2(\mathbf{x}_1 \to \mathbf{x}_2) \leq \sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}_1)^{1/2} P(\mathbf{y}|\mathbf{x}_2)^{1/2} \triangleq P_B(\mathbf{x}_1, \mathbf{x}_2), \tag{4.3.4}$$

defining $P_B(\mathbf{x}_1, \mathbf{x}_2)$. This is a rather general expression, not requiring a channel symmetry or memoryless behavior.

The bound in (4.3.4) is known as the ***Bhattacharyya bound*** on error probability, and its negative logarithm is known as the ***Bhattacharyya distance,*** $d_B(\mathbf{x}_1, \mathbf{x}_2)$, between two codewords or signal sequences:

$$d_B(\mathbf{x}_1, \mathbf{x}_2) = -\log[P_B(\mathbf{x}_1, \mathbf{x}_2)]. \tag{4.3.5}$$

Equivalently, the two-codeword upper bound on error probability is $P_B(\mathbf{x}_1, \mathbf{x}_2) = 2^{-d_B(\mathbf{x}_1, \mathbf{x}_2)}$.

Despite the two stages of bounding, we will find that $P_B(\mathbf{x}_1, \mathbf{x}_2)$ defined by (4.3.4) is surprisingly tight for most channels of interest. Note also the symmetry of this expression, for the subscripts 1 and 2 could be interchanged without changing the value of the sum in (4.3.4). It might seem that the *exact* error probabilities are also symmetric; that is, we are just as likely to confuse $\mathbf{x}_2$ for $\mathbf{x}_1$ as the reverse. However, this is not true on asymmetric channels (see Exercise 4.3.5). Nonetheless, the *bound* we have obtained is a symmetric bound for all DMCs.

This same bound can be interpreted as a Chernoff bound, as we now show. Given transmission of $\mathbf{x}_1$, an error occurs if $P(\mathbf{y}|\mathbf{x}_2) \geq P(\mathbf{y}|\mathbf{x}_1)$. (We shall be pessimistic regarding ties.) Equivalently, the error event is defined by the set of outcomes $\mathbf{y}$ for which $\log P(\mathbf{y}|\mathbf{x}_2) - \log P(\mathbf{y}|\mathbf{x}_1) \geq 0$. Thus, we are interested in

$$P(\mathbf{x}_1 \to \mathbf{x}_2) = P\left[\log \frac{P(\mathbf{y}|\mathbf{x}_2)}{P(\mathbf{y}|\mathbf{x}_1)} \geq 0 | \mathbf{x}_1 \text{ sent}\right]. \tag{4.3.6}$$

Let's define the log-likelihood ratio in (4.3.6) to be the random variable Z. By a Chernoff

bound argument, as in Section 2.4,

$$\begin{aligned} P(Z \geq 0|\mathbf{x}_1) &\leq e^{-s0} E_{Z|\mathbf{x}_1}[e^{sZ}] \\ &= E_{Y|\mathbf{x}_1} \exp\left[\frac{s \log P(\mathbf{y}|\mathbf{x}_2)}{P(\mathbf{y}|\mathbf{x}_1)}\right] \\ &= E_{Y|\mathbf{x}_1}\left[\frac{P(\mathbf{y}|\mathbf{x}_2)}{P(\mathbf{y}|\mathbf{x}_1)}\right]^s. \end{aligned} \tag{4.3.7}$$

(The cumbersome subscript on the expectation operator is to emphasize that the expectation is with respect to the variable Z or Y when conditioned on $\mathbf{x}_1$.)

The conditional expectation can be obtained by multiplying the quantity whose expectation is sought by $P(\mathbf{y}|\mathbf{x}_1)$ and summing over $\mathbf{y}$:

$$\begin{aligned} P(Z \geq 0|\mathbf{x}_1) &\leq \sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}_1)\left[\frac{P(\mathbf{y}|\mathbf{x}_2)}{P(\mathbf{y}|\mathbf{x}_1)}\right]^s \\ &= \sum_{\mathbf{y}} [P(\mathbf{y}|\mathbf{x}_1)]^{1-s}\left[P(\mathbf{y}|\mathbf{x}_2)\right]^s. \end{aligned} \tag{4.3.8}$$

We are interested in minimizing this expression with respect to $s > 0$. *If* the prescribed channel is symmetric and memoryless, then symmetry of (4.3.8) implies the minimum occurs when $s = \frac{1}{2}$. Substitution of $s = \frac{1}{2}$ into (4.3.8) then yields (4.3.4). For asymmetric channels and arbitrary choice of codewords, the general Chernoff formulation can be tighter when $s \neq \frac{1}{2}$.

The Bhattacharyya bound (or Chernoff bound) plays an important role in our subsequent analysis of coded communication systems (see for example [4], [5]). One appealing aspect of the Bhattacharyya distance is that it lends a partial geometric interpretation (see Figure 4.3.2) to general decision problems through d_B, generalizing the importance of Euclidean distance that we have already seen for the coherent Gaussian channel, or Hamming distance for the BSC. The Bhattacharyya distance possesses two of the usual

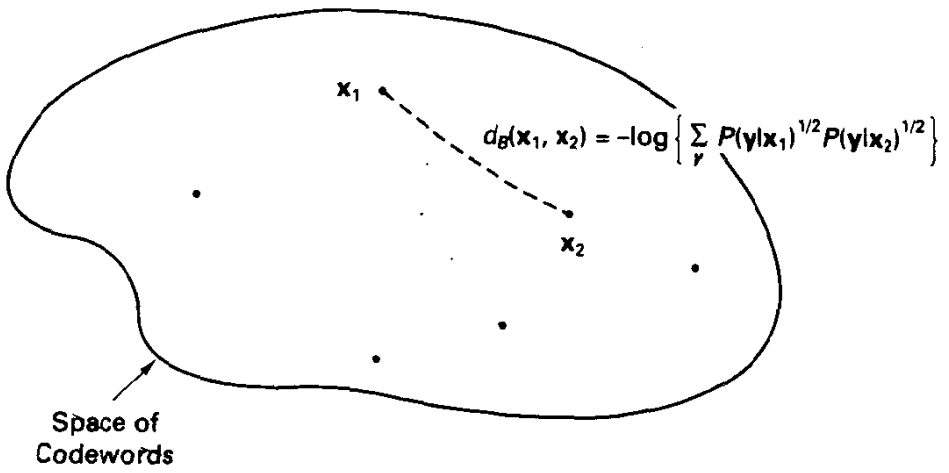

Figure 4.3.2 Bhattacharyya distance between two codewords.

attributes of a distance metric, nonnegativity and symmetry, but lacks the triangle inequality property (Exercise 4.3.3).

We next proceed to evaluate the Bhattacharyya bound (4.3.4). By the assumed memoryless property of the channel

$$P(\mathbf{y}|\mathbf{x}) = \prod_{j=0}^{n-1} P(y_j|x_j). \tag{4.3.9}$$

Substituting (4.3.9) into (4.3.4), expanding the n-fold sum, and recognizing that this can be written as a product of scalar summations, we find

$$P_B(\mathbf{x}_1, \mathbf{x}_2) = \prod_{j=0}^{n-1} \sum_{k=0}^{Q-1} \left[P(y_{kj}|x_{1j})P(y_{kj}|x_{2j})\right]^{1/2}, \tag{4.3.10a}$$

where Q is the size of the demodulator's output alphabet.

For memoryless channels producing a vector $\mathbf{y}_j$ of continuous random variables at each position, exactly the same line of reasoning may be followed as in (4.3.1) through (4.3.4), except that we replace summation over the output alphabet by integration and use conditional probability densities instead of probabilities. We obtain

$$P_B(\mathbf{x}_1, \mathbf{x}_2) = \prod_{j=0}^{n-1} \int_{-\infty}^{\infty} \left[f(\mathbf{y}_j|x_{1j})f(\mathbf{y}_j|x_{2j})\right]^{1/2} d\mathbf{y}_j. \tag{4.3.10b}$$

(The integral is a multiple integral over the space appropriate for $\mathbf{y}$.) Notice again that in both forms, (4.3.10a) and (4.3.10b), the resulting bound for error probability is a symmetric function of its two arguments.

The bound in (4.3.10) may be more compactly rewritten as

$$P_B(\mathbf{x}_1, \mathbf{x}_2) = \prod_{j=0}^{n-1} b_j \tag{4.3.11}$$

if we introduce the notation b_j for each term of the product in (4.3.10a) or (4.3.10b):

$$b_j = \sum_{k=0}^{Q-1} \left[P(y_k|x_{1j})P(y_k|x_{2j})\right]^{1/2}, \tag{4.3.12a}$$

or

$$b_j = \int_{-\infty}^{\infty} \left[f(\mathbf{y}_j|x_{1j})f(\mathbf{y}_j|x_{2j})\right]^{1/2} d\mathbf{y}_j. \tag{4.3.12b}$$

Notice that b_j is a function of the choice of two code symbols and the channel transition probabilities for the jth symbol position. Given two specific codewords, it is straightforward to evaluate (4.3.12) and hence (4.3.10), as the following examples illustrate.

Example 4.5 Two-codeword Bound on BSC

Suppose two binary codewords, (00000) and (10101), are to be used on a BSC with parameter ϵ, where ϵ is the error probability for each code symbol. (The fact that this is not the best choice of two codewords is immaterial.) In those positions of the codewords where the

symbols disagree (positions 1, 3, and 5), we can evaluate b_j from (4.3.12a) to be

$$\begin{aligned} b_j &= [P(0|0)P(0|1)]^{1/2} + [P(1|0)P(1|1)]^{1/2} \\ &= [\epsilon(1-\epsilon)]^{1/2} + [\epsilon(1-\epsilon)]^{1/2} \\ &= [4\epsilon(1-\epsilon)]^{1/2}. \end{aligned} \tag{4.3.13a}$$

In those positions where the code symbols agree, b_j is 1. Thus, we have

$$b_j = \begin{cases} [4\epsilon(1-\epsilon)]^{1/2}, & x_{1j} \neq x_{2j}, \\ 1, & x_{1j} = x_{2j}, \end{cases} \tag{4.3.13b}$$

and substitution into (4.3.11) yields

$$P_B(\mathbf{x}_1, \mathbf{x}_2) = [4\epsilon(1-\epsilon)]^{3/2} \tag{4.3.14}$$

since the codewords differ in three positions. More generally, we may write the Bhattacharyya upper bound on two-codeword error probability for the BSC as

$$P_B(\mathbf{x}_1, \mathbf{x}_2) = [4\epsilon(1-\epsilon)]^{d_H(\mathbf{x}_1, \mathbf{x}_2)/2}, \tag{4.3.15}$$

where $d_H(\mathbf{x}_1, \mathbf{x}_2)$ is the Hamming distance between the two codewords. We reemphasize that this same quantity serves as a bound on the probability that $\mathbf{x}_1$ would be chosen instead of $\mathbf{x}_2$, given the latter was selected for transmission.

In this simple case, it is easy to evaluate the *exact* probability of message error. A minimum Hamming distance decoder will fail if and only if two or more channel errors occur in the three positions where the codewords differ. (Notice that errors in the positions where the two codewords agree are not harmful.) Thus,

$$\begin{aligned} P(\mathbf{x}_1 \to \mathbf{x}_2) &= C_2^3 \epsilon^2 (1-\epsilon)^1 + C_3^3 \epsilon^3 \\ &= 3\epsilon^2(1-\epsilon) + \epsilon^3. \end{aligned} \tag{4.3.16}$$

As we would anticipate, the exact probability is strictly less than the Bhattacharyya bound, (4.3.14), for any $\epsilon < \frac{1}{2}$.

Example 4.6 Two-codeword Bound for Antipodal Signaling on AWGN Channel

Consider the case of binary coding where each binary code symbol is transmitted using an antipodal signal set. The basis function form of the demodulator produces scalar outputs that are Gaussian with mean either $E_s^{1/2}$ or $-E_s^{1/2}$ and variance $\sigma^2 = N_0/2$. The unquantized observation is passed to the decoder. From the integral form in (4.3.10b) we have that

$$b_j = \int_{-\infty}^{\infty} [f(y_j|x_{1j}) f(y_j|x_{2j})]^{1/2} \, dy_j. \tag{4.3.17}$$

Clearly, if $x_{1_j} = x_{2_j}$, then $b_j = 1$. If not, we can substitute the appropriate conditional density functions, expand, complete the square of the exponent, and then recognize the integral of a resulting p.d.f. to be 1. We find that

$$b_j = \begin{cases} e^{-E_s/N_0}, & x_{1j} \neq x_{2j}, \\ 1, & x_{1j} = x_{2j}. \end{cases} \tag{4.3.18}$$

(This is a special case of the general AWGN channel bound developed shortly, so we skip the details for the present.) Thus, for antipodal signaling in AWGN, the Bhattacharyya

bound becomes

$$P(\mathbf{x}_1 \to \mathbf{x}_2) \le P_B(\mathbf{x}_1, \mathbf{x}_2) = e^{-d_H(\mathbf{x}_1, \mathbf{x}_2)E_s/N_0}, \tag{4.3.19}$$

where again $d_H(\mathbf{x}_1, \mathbf{x}_2)$ is the Hamming distance between codewords.

It will be convenient to define the Bhattacharyya parameter B as the value of b_j when $x_{1j} \neq x_{2j}$. Thus, in Example 4.5, $B = [4\epsilon(1-\epsilon)]^{1/2}$, while in the present example the Bhattacharyya parameter is $B = e^{-E_s/N_0}$. In some sense, B will always depend on channel quality and we will have $B \le 1$.

Before proceeding with the development of the channel coding theorem, we detour briefly to consider **repetition coding**. Suppose that we have available a q-ary channel and exactly q messages, or codewords. Each codeword is formed by repeating *any* of the q symbols in the code alphabet n times, with obvious redundancy. The code rate is $R = (\log_2 q)/n$ bits/code symbol. In exchange for the rate per code symbol becoming small as n becomes large, we can, by (4.3.10) and (4.3.11), at least make the two-codeword error probability go to zero *exponentially* in n; that is,

$$P_2(\mathbf{x}_1 \to \mathbf{x}_2) \le B^n, \tag{4.3.20}$$

because the codewords differ in n positions. By a simple union bound, the probability of choosing one of $q-1$ incorrect codewords is less than $(q-1)B^n$, and thus we find that the probability of a message error is exponentially decreasing in n. We will soon find that vanishing throughput, or rate, need not be the price for high reliability.

4.3.1 Ensemble Average Performance for Two-codeword Codes

Now imagine that we do not focus on specific codewords $\mathbf{x}_1$ and $\mathbf{x}_2$, but that we form them by a probabilistic mechanism, with $P(\mathbf{x})$ denoting the probability assigned to n-tuples from an alphabet of size q. We assume that the two codewords are generated independently, and we also suppose that the code symbols of a given codeword are generated independently so that

$$P(\mathbf{x}_i) = \prod_{j=0}^{n-1} P(x_{ij}). \tag{4.3.21}$$

Thus, the scalar probability mass function $P(x)$ completely defines the probability structure for forming codewords. In a binary coding setting, we form codewords according to a coin-flipping process, not necessarily with a fair coin, however. The conceptual view is illustrated in Figure 4.3.3. This formulation is often described as a *random coding* strategy, although the name is misleading. Once we adopt a code, the coding process is completely deterministic, and the decoder is given the code in use.

A completely equivalent view is the following: We form an experiment that consists of random selection of a two-codeword code from the ensemble of *all* two-codeword codes of length n. The probability measure assigned to selection of a given code is just

$$P(\mathbf{x}_1, \mathbf{x}_2) = P(\mathbf{x}_1)P(\mathbf{x}_2) = \prod_{j=1}^{n} P(x_{1j})P(x_{2j}). \tag{4.3.22}$$

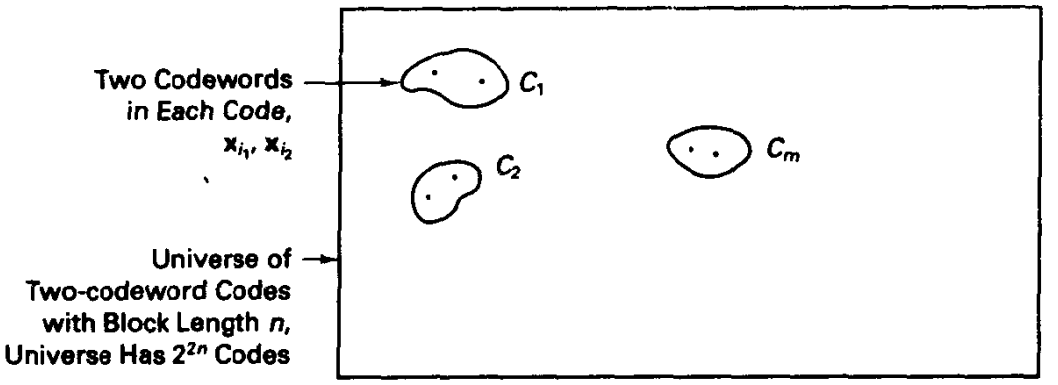

Figure 4.3.3 Ensemble of two-codeword block codes.

We now ask for an upper bound on the two-codeword error probability with a randomly selected pair of codewords. Equivalently, we could ask, "what is the probability that we randomly pick a code (with two codewords) and choose to send the first codeword, yet the decoder decides the second was sent?" We simply must average our previous result for two specific codewords. Noting that the two-codeword bound was symmetric in its arguments, we replace $P_2(\mathbf{x}_1 \to \mathbf{x}_2)$ by simply $P_2(\mathbf{x}_1, \mathbf{x}_2)$. Then the ensemble average error probability is

$$\overline{P_2(\mathbf{x}_1, \mathbf{x}_2)} \leq \sum_{\mathbf{x}_1} \sum_{\mathbf{x}_2} P(\mathbf{x}_1) P(\mathbf{x}_2) P_B(\mathbf{x}_1, \mathbf{x}_2) = \overline{P_B(\mathbf{x}_1, \mathbf{x}_2)}. \tag{4.3.23}$$

Substitution of the product distribution assumed for $P(\mathbf{x}_i)$ into (4.3.23) and using (4.3.10) gives, after manipulating sums,

$$\overline{P_2(\mathbf{x}_1, \mathbf{x}_2)} \leq \prod_{j=0}^{n-1} \sum_{y_j} \sum_{x_{1j}} \sum_{x_{2j}} P(x_{1j}) P(x_{2j}) \left[P(y_j|x_{1j}) P(y_j|x_{2j}) \right]^{1/2} \tag{4.3.24a}$$

After realizing that the subscripted variables in (4.3.24a) are merely dummy variables and that each term in the product is independent of position index j, we may simplify this result to

$$\overline{P_2(\mathbf{x}_1, \mathbf{x}_2)} \leq \prod_{j=0}^{n-1} \sum_{y} \left[\sum_{x} P(x) P(y|x)^{1/2} \right]^2, \tag{4.3.24b}$$

where the two summations are over the output and input alphabets, respectively.

To more compactly represent (4.3.24), we introduce a new quantity, $R_0(P)$:

$$R_0(P) = -\log_2 \left(\sum_{y} \left[\sum_{x} P(x) P(y|x)^{1/2} \right]^2 \right). \tag{4.3.25}$$

In this definition, $R_0(P)$ carries dimensions of bits/channel symbol. This definition allows writing the bound on error probability for the ensemble of two-codeword codes as

$$\overline{P_2(\mathbf{x}_i, \mathbf{x}_j)} \leq 2^{-nR_0(P)}. \tag{4.3.26}$$

Notice that we have switched now to general codeword subscripts, for the result would hold for any pair of codewords in a larger code, provided the codeword probability structure is unchanged.

We are free to choose the distribution on code symbols $P(x)$ so that we obtain the smallest upper bound. Thus, we define R_0 to be

$$R_0 = \max_{P(x)} \left\{ -\log_2 \left(\sum_y \left[\sum_x P(x)P(y|x)^{1/2} \right]^2 \right) \right\}, \tag{4.3.27a}$$

from which

$$\overline{P_2(\mathbf{x}_1, \mathbf{x}_2)} < 2^{-nR_0}. \tag{4.3.27b}$$

Due to monotonicity of the logarithm function, an equivalent expression for R_0 is

$$R_0 = \log_2 \left[\min_{P(x)} \left\{ \sum_y \left[\sum_x P(x)P(y|x)^{1/2} \right]^2 \right\} \right]. \tag{4.3.28}$$

For ***symmetric*** channels, as defined in Section 2.7, an equiprobable distribution on the input alphabet, $P(x) = 1/q$, achieves the extremum, as it does for capacity. This can be readily shown by study of (4.3.28). The expression for R_0 in this case becomes

$$\begin{aligned} R_0 &= -\log_2 \left(\sum_y \left[\frac{1}{q} \sum_x P(y|x)^{1/2} \right]^2 \right) \\ &= \log_2 q - \log_2 \left(\frac{1}{q} \sum_y \left[\sum_x P(y|x)^{1/2} \right]^2 \right). \end{aligned} \tag{4.3.29}$$

Even for channels that are not symmetric, R_0 given by (4.3.29) is a lower bound on (4.3.27a) and hence yields a valid upper bound for $\overline{P_B(\mathbf{x}_1, \mathbf{x}_2)}$. The second logarithm term on the right-hand side is positive, and thus $R_0 \leq \log_2 q$ bits per signaling interval, with equality approached as the channel quality improves.

Example 4.7 Application to the Binary Symmetric Channel

Let's choose two codewords of length $n = 10$ by tossing a fair coin. (We can imagine this as the choice of a code from the set of all possible codes with two codewords of length 10.) We may get an especially good code, say $\mathbf{x}_1 = (0000000000)$ and $\mathbf{x}_2 = (1111111111)$, or we may unfortunately produce identical codewords! Such is the spirit of "random coding." In any case, the expected error probability, with expectation taken with respect to choice of codes, is

$$\overline{P_2(\mathbf{x}_1, \mathbf{x}_2)} \leq 2^{-10R_0}, \tag{4.3.30}$$

where $R_0 = 1 - \log[1 + \sqrt{4\epsilon(1-\epsilon)}]$ as found from (4.3.29) and (4.3.27b). If $\epsilon = 0.1$, then $R_0 = 0.322$ bit per code symbol, and $\overline{P_B(x_1, x_2)} \leq 2^{-10(0.322)} = 0.108$. This is undoubtedly disappointing, since the expected error probability of a randomly selected code is slightly worse (at least by this bound) than obtained if we merely send uncoded messages 0 or 1 through the channel with one transmission!

We might compare this with the performance of the *best* previous code, whose ML decoder, using a majority vote, will decide correctly if fewer than five errors occur in a codeword transmission. If five errors occur, we choose either codeword with probability $\frac{1}{2}$.

The probability of a decoding error for such a code is

$$P(\mathbf{x}_1 \to \mathbf{x}_2) = \frac{1}{2}P(5 \text{ errors in } 10 \text{ transmissions}) + P(6 \text{ or more errors})$$
$$= \frac{1}{2}C_5^{10}\epsilon^5(1-\epsilon)^5 + \sum_{j=6}^{10} C_j^{10}\epsilon^j(1-\epsilon)^{10-j}, \tag{4.3.31}$$

which is $8.9 \cdot 10^{-4}$ for $\epsilon = 0.1$, clearly much smaller than our bound for a randomly selected pair of codewords. The bound's weakness derives from two sources: the inclusion of bad codes in the entire ensemble and the Bhattacharyya upper bound for a specific code. The example may suggest that random coding ideas are rather impotent, but in fact they are at the heart of the proofs of coding theorems to follow.

Historical Aside: The parameter R_0 initially surfaced in conjunction with sequential decoding of convolutional codes [6, 7], *where it has an important complexity implication. There it was called R_{comp}, for computational cutoff rate, for it was shown that attempts to transmit with code rates larger than R_{comp} were faced with a mean decoder computation per bit that was not finite. This has promulgated a folklore that, although we can in principle communicate at rate near channel capacity with arbitrarily small error probability, R_0 represents an upper limit on rate for practically instrumentable reliable communications. There seems to be little other precise support for this notion. Sequential decoding will be described in the context of trellis codes in Chapter 6.*

To recap, we have bounded the probability of error for two specific codewords of length n, in terms of the channel transition probability assignments, and called this the Bhattacharyya bound. We then proceeded to choose the codewords according to a probabilistic mechanism and found an upper bound on error probability for the ensemble of two-codeword codes. By definition, this quantity is 2^{-nR_0}.

4.3.2 Extension to Discrete-Input, Continuous-Output Channels

The previous development may be extended to the *discrete-input, continuous-output channel* as follows. We begin by assuming that the output of the channel for *each* code symbol is an N-dimensional continuous vector $\mathbf{y}$. For a specific choice of two codewords $\mathbf{x}_1$ and $\mathbf{x}_2$, we recall from (4.3.10b) that

$$P_B(\mathbf{x}_1, \mathbf{x}_2) = \prod_{j=0}^{n-1} \int_{-\infty}^{\infty} [f(\mathbf{y}_j|x_{1j}) f(\mathbf{y}_j|x_{2j})]^{1/2}\, d\mathbf{y}_j.$$

We again consider an ensemble of two-codeword codes and let the probability assignment on code symbols have the independent structure used in the discrete case. We let $P(x)$ denote the marginal probability assigned to each letter of each codeword. Formulation

of the ensemble average $\overline{P_2(\mathbf{x}_1, \mathbf{x}_2)}$ yields

$$\overline{P_2(\mathbf{x}_1, \mathbf{x}_2)} \leq 2^{-nR_0}, \tag{4.3.32a}$$

where

$$R_0 = \max_{P(x)} R_0(P) \tag{4.3.32b}$$

and where

$$R_0 = \max_{P(x)} \left\{ -\log_2 \int_{\mathbf{y}} \left(\sum_{x} P(x) f(\mathbf{y}|x)^{1/2} \right)^2 d\mathbf{y} \right\}. \tag{4.3.32c}$$

The integral in (4.3.32c) is interpreted as N-dimensional.

A helpful alternative argument is to imagine a discretized version of the output vector obtained by uniformly partitioning N-dimensional space into hypercubes. This produces a discrete memoryless channel to which (4.3.27) can be applied. The appropriate conditional probabilities would be obtained by integrating conditional density functions over the various regions. In the limit as the partition becomes fine, the sum over the output variable becomes an integral, and we obtain (4.3.32).

In an important special case, (4.3.32) can be considerably simplified. That is, consider an N-dimensional set of M signals to be transmitted over an AWGN channel with coherent detection. The receiver first projects the received waveform into signal space, obtaining the output vector $\mathbf{y} = (y_0, y_1, \ldots, y_{N-1})$. These are independent Gaussian random variables, each with variance $\sigma^2 = N_0/2$. Furthermore, the mean vector, conditioned on transmission of signal $s_i(t)$, is $\mathbf{s}_i = (s_{i0}, s_{i1}, \ldots, s_{i,N-1})$, the vector of signal-space coordinates. We shall utilize vector norm notation for manipulating densities; specifically, $\|\mathbf{y} - \mathbf{x}_i\|^2$ will be the squared Euclidean distance between $\mathbf{y}$ and $\mathbf{x}_i$. This can be expanded as

$$\|\mathbf{y} - \mathbf{x}_i\|^2 = \|\mathbf{y}\|^2 - 2\mathbf{y} \cdot \mathbf{x}_i + \|\mathbf{x}_i\|^2. \tag{4.3.33}$$

By substituting the known conditional density information into (4.3.32c) and realizing that the integrand is a product of sums, we obtain

$$\begin{aligned} R_0(P) = -\log \sum_{\mathbf{s}_1} \sum_{\mathbf{s}_2} \int & \left[\frac{1}{(2\pi\sigma^2)^{N/2}} e^{-(\|\mathbf{y}-\mathbf{s}_1\|^2/2\sigma^2)} \right]^{1/2} \\ & \left[\frac{1}{(2\pi\sigma^2)^{N/2}} e^{-(\|\mathbf{y}-\mathbf{s}_2\|^2/2\sigma^2)} \right]^{1/2} P(\mathbf{s}_1) P(\mathbf{s}_2)\, d\mathbf{y}. \end{aligned} \tag{4.3.34}$$

After expanding the integrand, we obtain

$$\begin{aligned} R_0(P) = -\log \sum_{\mathbf{s}_1} \sum_{\mathbf{s}_2} \int & \frac{1}{(2\pi\sigma^2)^{N/2}} \\ & \cdot e^{-(\|\mathbf{y}\|^2 - (\mathbf{s}_1+\mathbf{s}_2)\cdot\mathbf{y} + \|\mathbf{s}_1\|^2 + \|\mathbf{s}_2\|^2)/2\sigma^2} P(\mathbf{s}_1) P(\mathbf{s}_2)\, d\mathbf{y}. \end{aligned} \tag{4.3.35}$$

Completion of the square in the exponent yields

$$R_0(P) = -\log \sum_{\mathbf{s}_1} \sum_{\mathbf{s}_2} \int \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left[-\left\|\mathbf{y} - \frac{\mathbf{s}_1 + \mathbf{s}_2}{2}\right\|^2 / 2\sigma^2\right] d\mathbf{y}$$
$$\cdot \exp\left[-\left(\left\|\frac{\mathbf{s}_1 - \mathbf{s}_2}{2}\right\|^2 / 2\sigma^2\right)\right] P(\mathbf{s}_1)P(\mathbf{s}_2). \tag{4.3.36}$$

The integral is 1 for any choice of code vectors, since it is recognized as the integral of a multidimensional Gaussian p.d.f. Hence, the final result simplifies to

$$R_0(P) = -\log \sum_{\mathbf{s}_1} \sum_{\mathbf{s}_2} e^{-\|\mathbf{s}_1 - \mathbf{s}_2\|^2/4N_0} P(\mathbf{s}_1)P(\mathbf{s}_2), \tag{4.3.37}$$

where we have used $\sigma^2 = N_0/2$. This expression may be readily evaluated in terms of signal-space coordinates for *any* signal set, whether QAM, PSK, orthogonal, lattice-type, or other. For signal constellations where the set of distances to other neighbors is invariant to choice of reference point (a symmetry condition holding for M-PSK and M-orthogonal sets, for example), we have that the equiprobable assignment on inputs maximizes $R_0(P)$, and

$$R_0 = \log_2 M - \log_2 \left\{ \sum_{j=0}^{M-1} e^{-(\|\mathbf{s}_0 - \mathbf{s}_j\|^2/4N_0)} \right\}. \tag{4.3.38}$$

Even when the symmetry indicated previously is lacking, it is convenient to define R_0 according to (4.3.38), remembering that the exact value might be slightly superior.

In any case, R_0 approaches $\log_2 q$ bits/channel symbol as the energy-to-noise density ratio increases. This implies, by (4.3.32a), that for a randomly-selected code $P_2(\mathbf{x}_1, \mathbf{x}_2) < 2^{-nR_0} < q^{-n}$. The latter is just the probability that two codewords are equal in all positions, clearly an unfortunate choice of codewords.

In this section, we have demonstrated the significance of R_0 for the two-codeword situation. Obviously, our real interest is in the case where we have many (2^{nR}) codewords, which we pursue in the next section. We remark, however, that R_0 will play an important role in this case as well.

Example 4.7 Continued

As an application of (4.3.38), we suppose that the BSC assumed in Example 4.7 arose from use of binary antipodal signaling on an AWGN channel. The error probability assumed there, $\epsilon = 0.1$, corresponds to $E_s/N_0 = 0.817 = -0.87$ dB. Suppose that instead of making a binary decision on each coded symbol we retain the single Gaussian r.v. produced in the demodulator as the sufficient statistic. Decoding would then employ analog correlation in the likelihood computation, as we have earlier discussed. We thereby have a discrete-input, continuous-output communication channel.

Substitution in (4.3.38) gives

$$R_0 = \log_2 2 - \log_2(1 + e^{-E_s/N_0}) = 0.472 \text{ bit/channel use} \tag{4.3.39}$$

which we note is larger than the 0.322 bit/channel use for the corresponding quantized channel. Continuing the assumptions made earlier, we suppose that the block length of codewords is $n = 10$. Then the probability of error for two-codeword codes, averaged over the ensemble of all such codes, is

$$\overline{P(\mathbf{x}_1 \to \mathbf{x}_2)} = 2^{-nR_0} = 2^{-4.72} = 0.0379, \tag{4.3.40}$$

again significantly smaller than the corresponding result for the previous quantized channel. This is evidence for the deleterious effect of coarse receiver quantization in processing of specific codes.

Of course, the best code with two codewords performs far better. It corresponds to a repetition code, equivalent in signal-space terms to an antipodal signal set with E_b/N_0 of $10(0.817) = 8.17$. Optimal decoding will have a bit error probability given by the bit error probability for antipodal signals:

$$P_b = Q\left[(2 \cdot 8.17)^{1/2}\right] = 2.7 \cdot 10^{-5}. \tag{4.3.41}$$

4.3.3 Generalizations

This bounding procedure can be extended to allow for use of an arbitrary (not necessarily ML) metric on a memoryless channel. The channel inputs are q-ary, and we assume that the channel output is either discrete or continuous. Furthermore, we assume that the demodulator may supply side information for each time interval, such as channel amplitude or instantaneous noise level in a time-varying interference situation. We let the side information in interval n be represented by the variable(s) z_n.

We assume that a per-symbol metric $\lambda(x_n, \mathbf{y}_n; z_n)$ is employed for scoring the goodness of a given channel output vector $\mathbf{y}$ against a hypothesized code symbol. Some examples might be

$$\begin{aligned}
&\lambda(x_n, y_n; z_n) = -z_n d_H(x_n, y_n), \quad \text{weighted Hamming metric,} \\
&\lambda(x_n, y_n; z_n) = x_n y_n, \quad \text{AWGN metric,} \\
&\lambda(x_n, y_n; z_n) = z_n x_n y_n, \quad \text{amplitude-weighted correlation,} \\
&\lambda(x_n, \mathbf{y}_n; z_n) = y_{x_n}^2, \quad \text{square of correlator output for hypothesized signal.}
\end{aligned} \tag{4.3.42}$$

The last was encountered in Section 3.8 as an approximation to optimal combining of noncoherently detected frequency-hopping demodulator outputs.

The decoder will decide in favor of the codeword having the greatest codeword metric, which we assume to be a sum of symbol metrics:

$$\Lambda(\tilde{\mathbf{y}}, \mathbf{x}_i; \mathbf{z}) = \sum_{j=0}^{N-1} \lambda(\mathbf{y}_j, x_j; z_j). \tag{4.3.43}$$

Thus, the probability that $\mathbf{x}_2$ is selected when $\mathbf{x}_1$ is in fact sent is

$$P(\mathbf{x}_1 \to \mathbf{x}_2) = P(\Lambda(\tilde{\mathbf{y}}, \mathbf{x}_2; \mathbf{z}) - \Lambda(\tilde{\mathbf{y}}, \mathbf{x}_1; \mathbf{z}) > 0 | \mathbf{x}_1 \text{ sent}).$$

This probability is difficult to evaluate in the general case, but can be upper-bounded by a Chernoff bound. Thus, we have, for any $s > 0$,

$$\begin{aligned}
P(\mathbf{x}_1 \to \mathbf{x}_2 | \mathbf{x}_1) &\le E\left[e^{s(\Lambda(\tilde{\mathbf{y}}, \mathbf{x}_2; \mathbf{z}) - \Lambda(\tilde{\mathbf{y}}, \mathbf{x}_1; \mathbf{z}))} \,\middle|\, \mathbf{x}_1 \right] \\
&= E\left[\prod_{j=0}^{n-1} e^{s(\lambda(\mathbf{y}_j, x_{2j}; z_j) - \lambda(\mathbf{y}_j, x_{1j}; z_j))} \,\middle|\, \mathbf{x}_1 \right] \\
&= \prod_{j=0}^{n-1} E[e^{s(\lambda(\mathbf{y}_j, x_{2j}; z_j) - \lambda(\mathbf{y}_j, x_{1j}; z_j))} | x_{ij}].
\end{aligned} \tag{4.3.44}$$

This expectation is with respect to choice of the code symbols in the various positions and with respect to the channel action.

Clearly, when two code symbols agree in a given position, the contribution to the product is a factor 1. For channels/metrics having ***output symmetry***,[2] when $x_{1j} \neq x_{2j}$, the jth factor is independent of the specific values x_{1j}, x_{2j}.

In this symmetric case, defining

$$D(s) = E\left[e^{s(\lambda(y_j, x_{2j}; z_j) - \lambda(y_j, x_{1j}; z_j))} | x_{1j}\right] \tag{4.3.45}$$

allows us to write

$$P(\mathbf{x}_1 \to \mathbf{x}_2) \leq D(s)^{d_H(\mathbf{x}_1, \mathbf{x}_2)}, \tag{4.3.46}$$

and by defining $D = \min D(s)$, we obtain the tightest upper bound:

$$P(\mathbf{x}_1 \to \mathbf{x}_2) \leq D^{d_H(\mathbf{x}_1, \mathbf{x}_2)}. \tag{4.3.47}$$

This illustrates the importance of Hamming distance in general transmission settings.

4.4 PROBABILITY OF ERROR WITH MANY CODEWORDS AND THE CHANNEL CODING THEOREM

Using the methods of the previous section, we now move to the case of codes with many codewords and develop expressions for the performance of ensembles of codes. The first argument, which is simple but not the strongest available, is a union-bound argument that endows the parameter R_0 with a twofold significance. The second argument, due to Gallager [8], is more subtle and establishes the fundamental noisy channel coding theorem and the role of channel capacity C for a DMC. R_0 emerges in this development as well.

4.4.1 Code Ensembles and a Simple Ensemble Bound on Performance

Imagine the universe of all possible block codes with rate R, block length n, and alphabet size q. Each code has $T = 2^{nR}$ codewords of length n symbols. Each of the $n2^{nR}$ symbols can be any of q choices, so there are $q^{n2^{nR}}$ possible codes. Even for modest-sized codes with $q = 2$, $n = 10$, and $R = 0.5$, where each code would have 32 codewords of length 10, the number of distinct codes is 2^{320}. Some of these codes are quite poor, for the ensemble includes q^n different code sets whose codewords are all identical and many more codes with at least one pair of duplicate codewords. While it is easy to describe poor codes, it is a difficult problem to describe the best code in the ensemble, let alone evaluate its performance exactly. So, after Shannon, we take a different route. We are interested in evaluating the *message error probability averaged over the ensemble of codes*. Despite the presence of weak codes just described, the ensemble average

[2]This includes antipodal modulation with the correlation metric and orthogonal signaling with square-law metric.

performance can be impressive as we shall see, provided $R < C$, thereby convincing us that at least one code in the ensemble must be good.

Consider the following thought experiment, with reference to Figure 4.4.1. A message source selects a codeword index i, $i = 1, 2, \ldots, T$. At the same time we perform a code selection experiment: we imagine selection of a code from the code universe, according to a probability distribution that is of a very simple form. We assume that codeword probabilities $P(\mathbf{x}_j)$ are independent, and symbols within a codeword are also independent, as in Section 4.3. An alternative way of thinking about this process is to form all the codewords in our code by a sequence of independent trials of a q-ary experiment, outcomes not necessarily equiprobable. This is the random coding framework of the previous section.

A specific code C, when used to send the message labeled i through a DMC, will exhibit some error probability $P(e|i, C)$, which in general depends on the code selected as well as the message index. More specifically, we are interested in the probability that we send a codeword, say $\mathbf{x}_i$, and find that some other codeword $\mathbf{x}_j$ in the code has equal or higher likelihood, causing a decoding error. (We shall be pessimistic about resolving likelihood ties.) The error event $\{\hat{\mathbf{x}}_i \neq \mathbf{x}_i\}$ is the union of $T - 1$ error events of specific type, and we can apply a union bound to express the conditional error probability for a specific code C as

$$\begin{aligned} P(e|i, C) &\leq \sum_{j \neq i} P(\mathbf{x}_i \to \mathbf{x}_j | C) \\ &\leq \sum_{j \neq i} P_B(\mathbf{x}_i, \mathbf{x}_j | C), \end{aligned} \tag{4.4.1}$$

again invoking the Bhattacharyya (or Chernoff) bound of the previous section, but leaving the code conditioning in force.

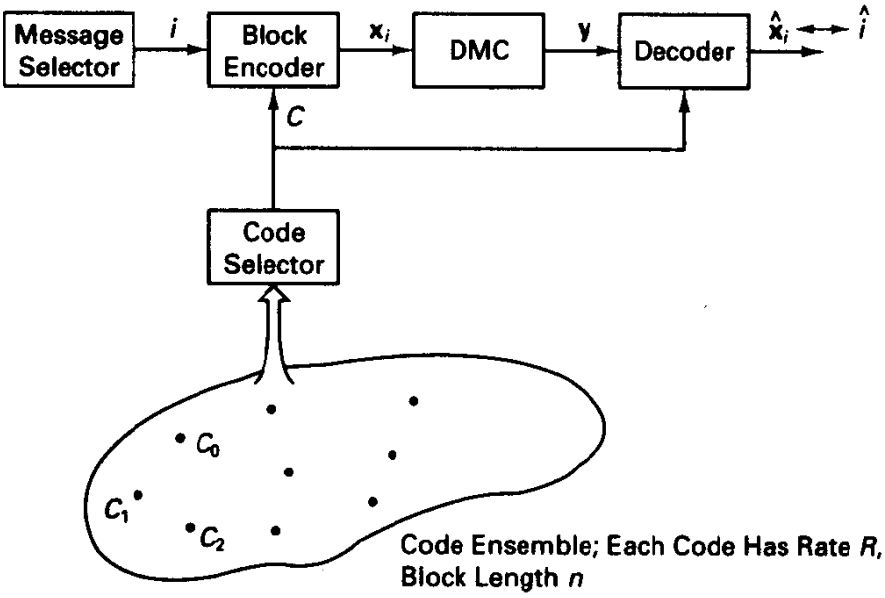

Figure 4.4.1 Framework for "random coding."

Next, we compute the ensemble average (with respect to code selection, keeping i fixed) of (4.4.1) and obtain

$$\overline{P(e|i)} \leq \sum_{j \neq i} \overline{P_B(\mathbf{x}_i, \mathbf{x}_j)}. \tag{4.4.2}$$

Each ensemble average in the sum of (4.4.2) is equivalent and independent of i and j. Thus, we find that the ensemble of codes has an error probability, for any message index i, bounded by

$$\overline{P(e)} \leq (T-1)\overline{P_B(\mathbf{x}_i, \mathbf{x}_j)} = (T-1)2^{-nR_0}, \tag{4.4.3}$$

invoking the definition of R_0 from the last section. Now upper-bounding $T - 1$ by $T = 2^{nR}$, we have that the ensemble average error probability is bounded by

$$\overline{P(e)} < 2^{nR}2^{-nR_0} = 2^{-n(R_0-R)}, \tag{4.4.4}$$

which, provided $R < R_0$, affords exponentially decreasing error probability for the ensemble average as block length increases. This is sometimes known as a ***union-Chernoff bound*** or ***union-Bhattacharyya bound.***

We will soon find it convenient to write (4.4.4) as

$$\overline{P(e)} \leq 2^{-nE(R)}, \tag{4.4.5}$$

where $E(R) = R_0 - R$ is an *error exponent* that is a function of code rate R. This error exponent is positive provided the code rate is less than the channel parameter R_0 (see Figure 4.4.2). Moreover, for any rate R, $R_0 - R$ specifies the size of the exponent. Thus, we see, based on random coding arguments, that R_0 establishes both a range of rates where reliable communication (arbitrarily small message error probability) is possible

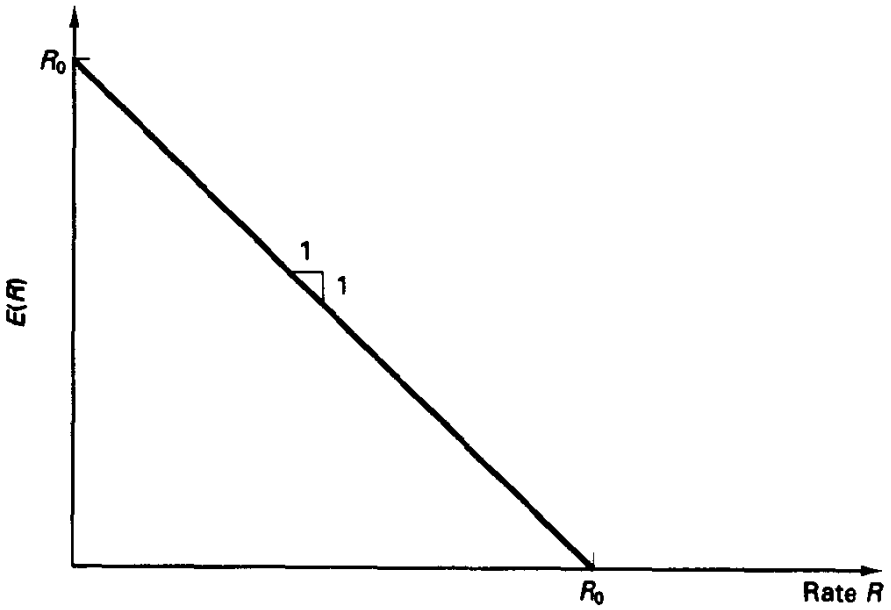

Figure 4.4.2 Error exponent $E(R)$ produced by union–Bhattacharyya bound. Both intercepts equal R_0.

and a forecast of the performance as well. This suggests that R_0 is a single-parameter descriptor of a channel's quality [2].

Because at least one code in the ensemble has performance matching or beating the ensemble average, we have demonstrated the *existence* of good codes as block length increases, at least for a range of rates less than R_0. Furthermore, R_0 provides some indication of the required block length needed to achieve a given performance target on a memoryless channel, but it is wise not to accept (4.4.4) too strictly in system designs.

Random coding is certainly not a recipe for finding easily instrumentable and powerful codes, but it is not as unsophisticated as it may seem. Markov's inequality implies that if the ensemble of codes of a given rate and block length has error probability $\overline{P(e)}$, then no more than 1% of the codes can have error probability larger than $100\overline{P(e)}$, and so on. Thus, the aphorism at the head of the chapter—in effect, randomly picked codes of appropriate rate are highly likely to be good in the sense of providing high reliability as blocklength increases; however, the lack of code structure would generally make encoding and decoding prohibitively difficult.

We will return later in this chapter to a thorough discussion of R_0 and its implications for modulation and coding design. However, we will first develop a stronger result, developed in principle by Shannon and extended by many others, that the range of rates for which exponentially decreasing error probability holds is $R < C$, with $C \geq R_0$. Attainment of this stronger result necessitates using something more clever than the union-Bhattacharyya bounding approach. Our development closely follows that of Gallager [8].

4.4.2 Generalized Upper Bound for a Specific Code with Many Codewords

We consider again the ensemble of codes of size T and block length n with codewords denoted by $\mathbf{x}_i, i = 1, \ldots, T$. Suppose we select for transmission the ith message. The codeword corresponding to this message is viewed as a codeword of a randomly selected code C, with the same independent probability model for codewords and codes used previously. Let $\mathbf{y}$ be the corresponding output vector produced by the DMC in response to $\mathbf{x}_i$. We denote the conditional error probability, given $i, \mathbf{x}_i$, and $\mathbf{y}$, by $P(e|i, \mathbf{x}_i, \mathbf{y})$. The error probability for the ith message, averaged over the code ensemble, is then

$$\begin{aligned}\overline{P(e|i)} &= \sum_{\mathbf{x}_i} \sum_{\mathbf{y}} P(e|i, \mathbf{x}_i, \mathbf{y}) P(\mathbf{x}_i, \mathbf{y}) \\ &= \sum_{\mathbf{x}_i} \sum_{\mathbf{y}} P(\mathbf{x}_i) P(\mathbf{y}|\mathbf{x}_i) P(e|i, \mathbf{x}_i, \mathbf{y}).\end{aligned} \tag{4.4.6}$$

Next, we proceed to upper-bound the conditional error probability term in (4.4.6). Given $\mathbf{x}_i$ and $\mathbf{y}$, an error will be made if *some* other codeword $\mathbf{x}_j$ in the code has equal or greater likelihood; that is, $P(\mathbf{y}|\mathbf{x}_j) \geq P(\mathbf{y}|\mathbf{x}_i)$ for some $j \neq i$. We denote by $A_{j,i}(\mathbf{y})$ the event that, given $\mathbf{y}$, the codeword $\mathbf{x}_j$ has likelihood greater than or equal that of $\mathbf{x}_i$.

Then

$$P(e|i, \mathbf{x}_i, \mathbf{y}) \leq P\left[\bigcup_{j \neq i} A_{ji}(\mathbf{y})\right], \tag{4.4.7}$$

where inequality is allowed because tie breaking may succeed.

Now we apply a ***generalized union bound*** to the right-hand side of (4.4.7):

$$P\left(\bigcup B_j\right) \leq \left[\sum_j P(B_j)\right]^\rho, \qquad 0 \leq \rho \leq 1. \tag{4.4.8}$$

This is an extension of the more familiar union bound of Chapter 2 [obtained in (4.4.8) when $\rho = 1$] and is easily demonstrated as follows. If the sum in (4.4.8) is less than 1, raising the sum to a power between 0 and 1 cannot decrease the result. Also, if the sum is 1 or perhaps larger due to event overlap, raising the sum to a nonnegative power cannot make the result less than 1, and the inequality thus holds trivially.

Now, from the definition of $A_{ji}(\mathbf{y})$, we have that

$$P[A_{ji}(\mathbf{y})] = \sum_{\mathbf{x}_j \in X_i'} P(\mathbf{x}_j), \tag{4.4.9}$$

where X_i' is defined as the set of codewords $\mathbf{x}_j$ for which $P(\mathbf{y}|\mathbf{x}_j) \geq P(\mathbf{y}|\mathbf{x}_i)$. The sum in (4.4.9) is n-dimensional. As in the two-codeword development, we multiply each term in (4.4.9) by a factor larger than or equal to 1. Here we choose as our multiplier $[P(\mathbf{y}|\mathbf{x}_j)/P(\mathbf{y}|\mathbf{x}_i)]^s$, $s > 0$, which satisfies our need for all $\mathbf{x}_j$ in the constraint set of (4.4.9). (We previously adopted $s = \frac{1}{2}$ in formulating the two-codeword bound.) Thus, (4.4.9) is bounded as

$$P[A_{ji}(\mathbf{y})] \leq \sum_{\mathbf{x}_j \in X_i'} P(\mathbf{x}_j) \left[\frac{P(\mathbf{y}|\mathbf{x}_j)}{P(\mathbf{y}|\mathbf{x}_i)}\right]^s. \tag{4.4.10}$$

Further relaxation of the bound by including all $\mathbf{x}_j$ in the range of summation gives

$$P[A_{ji}(\mathbf{y})] \leq \sum_{\mathbf{x}_j} P(\mathbf{x}_j) \left[\frac{P(\mathbf{y}|\mathbf{x}_j)}{P(\mathbf{y}|\mathbf{x}_i)}\right]^s. \tag{4.4.11}$$

Observe that in (4.4.11) $\mathbf{x}_j$ is merely a variable of summation, and we see the bound is independent of j. Furthermore, since there are $T - 1$ selections for $j \neq i$, we have, from (4.4.7) and (4.4.8),

$$P(e|i, \mathbf{x}_i, \mathbf{y}) \leq \left[(T-1)P(\mathbf{x})\left[\frac{P(\mathbf{y}|\mathbf{x})}{P(\mathbf{y}|\mathbf{x}_i)}\right]^s\right]^\rho, \qquad s > 0, \quad 0 \leq \rho \leq 1. \tag{4.4.12}$$

Substitution of this conditional error probability into (4.4.6) and rearrangement yields

$$\overline{P(e|i)} \leq (T-1)^\rho \sum_{\mathbf{y}} \left[\sum_{\mathbf{x}_i} P(\mathbf{x}_i) P(\mathbf{y}|\mathbf{x}_i)^{1-s\rho}\right]\left[\sum_{\mathbf{x}} P(\mathbf{x}) P(\mathbf{y}|\mathbf{x})^s\right]^\rho. \tag{4.4.13}$$

We now choose $s = (1 + \rho)^{-1}$, which is nonnegative as required for $0 \leq \rho \leq 1$. (This choice may be shown [8, 9] to minimize (4.4.13) with respect to s for a given ρ, but in

any case the bound is preserved by this choice.) This produces

$$\overline{P(e|i)} \leq (T-1)^{\rho} \sum_{\mathbf{y}} \left[\sum_{\mathbf{x}} P(\mathbf{x}) P(\mathbf{y}|\mathbf{x})^{1/(1+\rho)} \right]^{1+\rho}, \qquad 0 \leq \rho \leq 1. \tag{4.4.14}$$

This expression is a general bound on an ensemble average, holding for arbitrary discrete channels. Looking ahead, we anticipate minimizing this bound by suitable choice of ρ and the input distribution $P(\mathbf{x})$. Notice that this result reduces when $\rho = 1$ to our earlier result for the two-codeword ensemble error probability with $T = 2$.

For the case of a q-input, Q-output DMC with $P(\mathbf{x})$ having a product form, as we earlier assumed for the ensemble of codes, the n-fold summations in (4.4.14) may be rewritten as

$$\overline{P(e|i)} \leq (T-1)^{\rho} \left(\sum_{j=0}^{Q-1} \left[\sum_{k=0}^{q-1} P(k) P(j|k)^{1/(1+\rho)} \right]^{1+\rho} \right)^{n}. \tag{4.4.15}$$

[We have now adopted the shorthand notation $P(k)$ for $P(x_k)$ and $P(j|k)$ for $P(y_j|x_k)$.]

After noting that $T - 1 < 2^{nR}$, we obtain the expression

$$\overline{P(e|i)} < 2^{nR\rho} \left(\sum_{j=0}^{Q-1} \left[\sum_{k=0}^{q-1} P(k) P(j|k)^{1/(1+\rho)} \right]^{1+\rho} \right)^{n} \tag{4.4.16}$$

To obtain a final and more compact expression, we realize that we are free to choose ρ in the unit interval, as well as $P(k)$, the probability assignment on the symbols in the code alphabet, and do so to minimize the bound (4.4.16). Thus, we define the ***random coding exponent*** (also known as a reliability function) as

$$E(R) = \max_{P} \max_{\rho} [E_0(\rho, P) - \rho R], \qquad 0 \leq \rho \leq 1, \tag{4.4.17}$$

where

$$E_0(\rho, P) = -\log_2 \sum_{j=0}^{Q-1} \left[\sum_{k=0}^{q-1} P(k) P(j|k)^{1/(1+\rho)} \right]^{1+\rho} \tag{4.4.18}$$

is what is called the Gallager function [8].

With these definitions we have established that

$$\overline{P(e|i)} < 2^{-nE(R)}, \tag{4.4.19}$$

and since the bound is valid for each message index i in the code, (4.4.19) becomes an upper bound on the ensemble error probability, irrespective of message index i or the probabilities of selecting the various messages for transmission.

We have yet to show that the error exponent of (4.4.17) is positive for a given range of rates, that is, for $0 < R < C$, but assuming that $E(R) > 0$ for some rate R, we have demonstrated that the ensemble average error probability can be driven to zero exponentially fast by increasing the block length n. At least one code in this ensemble must be at least this good, certifying the existence of a sequence of codes of increasing block length, but with fixed rate R, whose error probability diminishes exponentially with block length. Of course, our argument has not revealed the detailed nature of these codes, nor have we required that there be any structure allowing possibly simple encoding and decoding.

Furthermore, the argument has not actually said there is a sequence of codes for which *all* message indexes in the code have diminishing error probability; this is taken up shortly.

4.4.3 Properties of the Error Exponent and a Coding Theorem

We shall now examine the random coding exponent defined in (4.4.17) and (4.4.18) and thereby prove a general coding theorem for the DMC. We first recall the mutual information shared between channel input and output, given an input distribution $P(k), k = 0, \ldots, q-1$, and a channel transition law $P(j|k)$ is

$$I_P(X;Y) = \sum_k \sum_j P(k)P(j|k) \log \left(\frac{P(j|k)}{\sum_i P(i)P(j|i)} \right). \tag{4.4.20}$$

The subscript on mutual information designates that the result depends on the choice of input probability assignment. We assume that $I_P(X;Y)$ is strictly positive, which will be true for all but useless channels or degenerate input distributions. The function $E_0(\rho, P)$ in (4.4.18) has three fundamental properties:

$$E_0(\rho, P) \geq 0, \qquad \rho \geq 0 \tag{4.4.21a}$$

$$I_P(X;Y) \geq \frac{\partial E_0(\rho, P)}{\partial \rho} > 0, \qquad \rho \geq 0 \tag{4.4.21b}$$

with equality on the left obtained when $\rho = 0$; and

$$\frac{\partial^2 E_0(\rho, P)}{\partial^2 \rho} \leq 0, \qquad \rho \geq 0. \tag{4.4.21c}$$

These properties may all be demonstrated by beginning with the definition of $E_0(\rho, P)$ and applying standard calculus. (Gallager [9, p. 142] provides a detailed development as well.) In words, $E_0(\rho, P)$ is a nonnegative convex $\cup$ function of ρ over the given range, with slope at $\rho = 0$ equaling the corresponding mutual information. Visualized graphically as a function of ρ, $E_0(\rho, P)$ is sketched in Figure 4.4.3a. We note that the graph depends on the input distribution P.

Now, according to (4.4.17), for any given P distribution, we wish to maximize $E_0(\rho, P) - \rho R$ for $0 \leq \rho \leq 1$ (see Fig. 4.4.3b). A stationary point, if it exists, will be the solution to

$$\frac{\partial E_0(\rho, P)}{\partial \rho} - R = 0. \tag{4.4.22}$$

[If a stationary point exists, it will be a maximum by (4.4.21c).] Such a solution will exist in the interval $0 \leq \rho \leq 1$ if (see Figure 4.4.3)

$$\left.\frac{\partial E_0(\rho, P)}{\partial \rho}\right|_{\rho=1} \leq R \leq \left.\frac{\partial E_0(\rho, P)}{\partial \rho}\right|_{\rho=0} \equiv I_P(X;Y). \tag{4.4.23}$$

On the other hand, if $R < R_{cr}(P) = \partial E_0(\rho, P)/\partial \rho|_{\rho=1}$,[3] then the maximizing choice for ρ is $\rho = 1$, and the error exponent, which remember still depends on choice of the

[3] R_{cr} stands for "critical rate," although the name gives it more significance than it deserves.

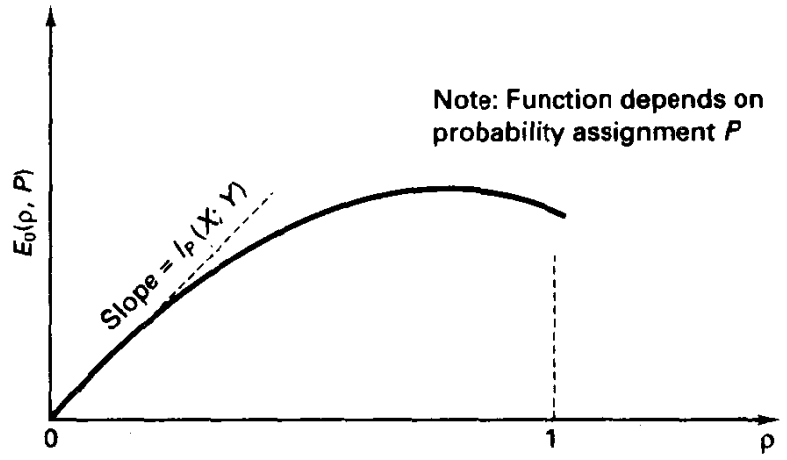

Figure 4.4.3a Gallager's $E_0(\rho, P)$ function.

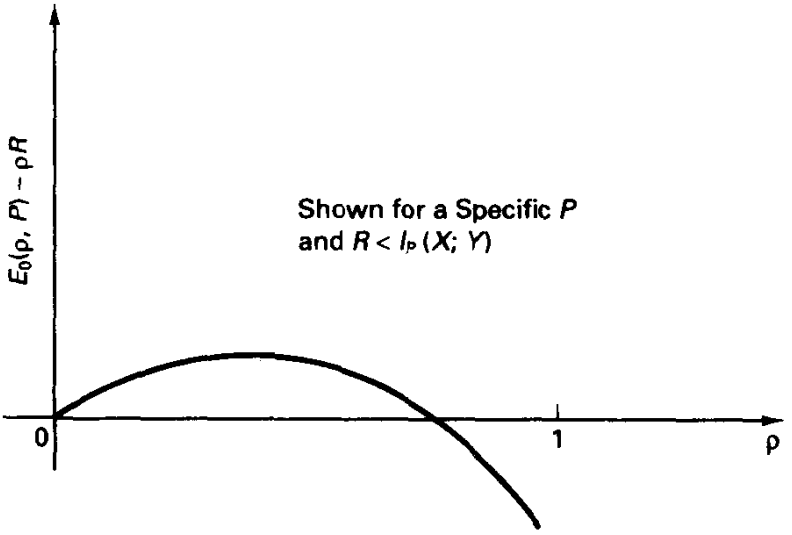

Figure 4.4.3b Function to be maximized over ρ for specific input distribution P.

input distribution P, becomes

$$E(R, P) = E_0(1, P) - R, \qquad R < R_{cr}(P). \tag{4.4.24}$$

Similarly, for R larger than the critical rate $R_{cr}(P)$, we have, using (4.4.22), a parametric form of the solution:

$$\begin{aligned} R &= \frac{\partial E_0(\rho, P)}{\partial \rho} \quad \text{(specifies } \rho\text{)} \\ E(R, P) &= E_0(\rho, P) - \rho \frac{\partial E_0(\rho, P)}{\partial \rho}, \end{aligned} \tag{4.4.25}$$

which pertains for $R_{cr}(P) \leq R < I_P(X; Y)$.

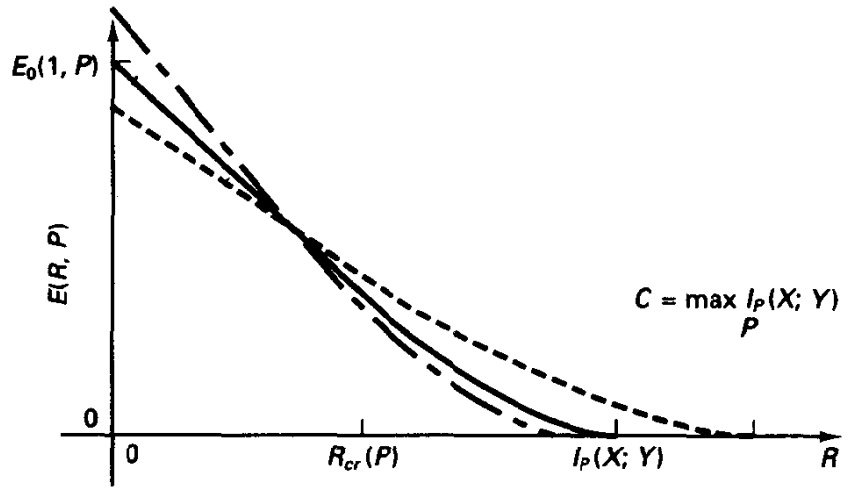

Figure 4.4.4 Error exponent $E(R)$ is upper envelope of $E(R, P)$ curves.

A typical sketch of $E(R, P)$ versus code rate R is shown in Figure 4.4.4, and we emphasize that its graph depends on the choice of P. For a given P, however, it may be shown that $E(R, P)$ is a positive, nonincreasing, convex $\cup$ function of R for $0 < R < I_P(X; Y)$.

Recalling the definition of the error exponent, $E(R)$, from (4.4.17), we now wish to maximize $E(R, P)$ over choice of input distributions P. The error exponent $E(R)$ is then visualized as the upper envelope of the family of all curves $E(R, P)$, as shown in Figure 4.4.4. It is obvious that for all rates $0 < R < \max I(X; Y) \equiv C$, $E(R)$ remains positive, which is the essential result. From Figure 4.4.4 we observe that, in general, a different input distribution P optimizes $E(R)$ as code rate R changes. However, *for symmetric channels*, as with the attainment of capacity, the equiprobable input assignment maximizes $E(R)$ for any rate R.

The general shape of $E(R)$ appears in Figure 4.4.5. For symmetric channels, there exists a straight-line portion of the curve, where $R < R_{cr}$, and in this range of rates, $E(R) = b - R$, where b is an intercept. This zero-rate intercept (see Figure 4.4.5) of the function $E(R)$ is in fact

$$\begin{aligned} E(R)|_{R=0} &= \max_P \max_\rho E_0(\rho, P) - \rho R|_{R=0} \\ &= \max_P \max_\rho E_0(\rho, P) = \max_P E_0(1, P) \end{aligned} \tag{4.4.26}$$

since $E_0(\rho, P)$ maximizes at $\rho = 1$. This latter term is

$$\max_P E_0(1, P) = \max_P \left\{ -\log \left[\sum_{j=0}^{Q-1} \left[\sum_{k=0}^{q-1} P(k) P(j|k)^{1/2} \right]^2 \right] \right\} \triangleq R_0. \tag{4.4.27}$$

Thus, the R_0 parameter emerges again as a key parameter describing the general random coding exponent: it is the zero-rate intercept of the random coding error exponent $E(R)$ derived previously, and in the low-rate region, it specifies an ensemble average upper-bound exponent.