Chapter 1

INTRODUCTION

§1.1. Why You Should Read This Book

The technology of communication and computing advanced at a breathtaking pace in the 20th century, especially in the second half. A significant part of this advance in communication began some 60 years ago when Shannon published his seminal paper "A Mathematical Theory of Communication." In that paper Shannon framed and posed a fundamental question: how can we efficiently and reliably transmit information? Shannon also gave a basic answer: coding can do it. Since that time the problem of finding practical coding schemes that approach the fundamental limits established by Shannon has been at the heart of information theory and communications. Recently, significant advances have taken place that bring us close to answering this question. Perhaps, at least in a practical sense, the question has been answered. This book is about that answer.

The advance came with a fundamental paradigm shift in the area of coding that took place in the early 1990s. In *Modern Coding Theory*, codes are viewed as large complex systems described by *random sparse graphical models*, and encoding as well as decoding are accomplished by efficient *local* algorithms. The local interactions of the codebits are simple but the overall code is nevertheless complex (and so sufficiently powerful to allow reliable communication) because of the large number of interactions. The idea of random codes is in the spirit of Shannon's original formulation. What is new is the sparseness of the description and the local nature of the algorithms.

These are exciting times for coding theorists and practitioners. Despite all the progress made, many fundamental questions are still open. Even if you are not interested in coding itself, however, you might be motivated to read this book. Although the focus of this book is squarely on coding, the larger view holds a much bigger picture. Sparse graphical models and message-passing algorithms, to name just two of the notions that are fundamental to our treatment, play an increasingly important role in many other fields as well. This is not a coincidence. Many of the innovations were brought into the field of coding by physicists or computer scientists. Conversely, the success of modern coding has inspired work in several other fields.

Modern coding will not displace classical coding anytime soon. At any point in time hundreds of millions of Reed-Solomon codes work hard to make your life less error prone. This is unlikely to change substantially in the near future. But mod-

ern coding offers an alternative way of solving the communications problem. Most current wireless communications systems have already adopted modern coding.

Technically, our aim is focused on Shannon's classical problem: we want to transmit a *message* across a *noisy channel* so that the *receiver* can determine this message with *high probability* despite the imperfections of the channel. We are interested in *low-complexity* schemes that introduce *little delay* and allow *reliable* transmission close to the ultimate limit, the *Shannon capacity*.

We start with a review of the communications problem (Section 1.2), we cover some classical notions of codes (Sections 1.3, 1.4, 1.5, 1.7, and 1.8), and we review the channel coding theorem (Section 1.6). Section 1.9 gives an outline of the modern approach to coding. Finally, we close in Section 1.10 with a review of the notational conventions and some useful facts.

§1.2. Communications Problem

Consider the following communications scenario – the *point-to-point* communications problem depicted in Figure 1.1. A *source* transmits its information (speech, audio, data, etc.) via a noisy channel (phone line, optical link, wireless, storage medium, etc.) to a *sink*. We are interested in reliable transmission, i.e., we want to recreate the transmitted information with as little *distortion* (number of wrong bits, mean squared error distortion, etc.) as possible at the sink.

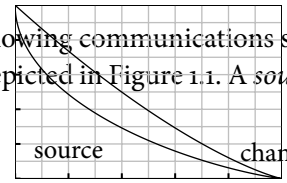

Figure 1.1: Basic point-to-point communications problem.

In his seminal paper in 1948, Shannon formalized the communications problem and showed that the point-to-point problem can be decomposed into two separate problems as shown in Figure 1.2. First, a *source encoder* transforms the source into a bit stream. Ideally, the source encoder removes all redundancy from the source so that the resulting bit stream has the smallest possible number of bits while still representing the source with enough accuracy. The *channel encoder* then processes the bit stream to add redundancy. This redundancy is carefully chosen to combat the noise that is introduced by the channel.

To be mathematically more precise: we model the output of the source as a stochastic process. For example, we might represent text as the output of a Markov chain, describing the local dependency structure of letter sequences. It is the task of the source encoder to represent this output as efficiently as possible (using as few bits as possible) given a desired distortion. The *distortion measure* reflects the "cost" of deviating from the original source output. If the source emits points in $\mathbb{R}^n$ it might

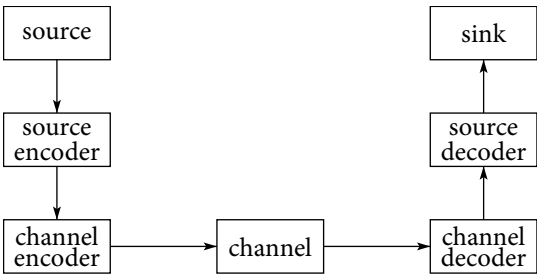

Figure 1.2: Basic point-to-point communications problem in view of the source-channel separation theorem.

be natural to consider the squared Euclidean distance, whereas if the source emits binary strings a more natural measure might be to count the number of positions in which the source output and the word that can be reconstructed from the encoded source differ. Shannon's *source coding theorem* asserts that, for a given source and distortion measure, there exists a minimum rate $R = R(d)$ (bits per emitted source symbol) which is necessary (and sufficient) to describe this source with distortion not exceeding d. The plot of this rate R as a function of the distortion d is usually called the *rate-distortion* curve. In the second stage an appropriate amount of *redundancy* is added to these source bits to protect them against the errors in the channel. This process is called *channel coding*. Throughout the book we model the channel as a probabilistic mapping and we are typically interested in the *average* performance, where the average is taken over all channel realizations. Shannon's *channel coding theorem* asserts the existence of a maximum rate (bits per channel use) at which information can be transmitted reliably, i.e., with vanishing probability of error, over a given channel. This maximum rate is called the *capacity* of the channel and is denoted by C. At the receiver we first decode the received bits to determine the transmitted information. We then use the decoded bits to reconstruct the source at the receiver. Shannon's *source-channel separation* theorem asserts that the source can be reconstructed with a distortion of at most d at the receiver if $R(d) < \mathrm{C}$, i.e., if the rate required to represent the given source with the allowed distortion is smaller than the capacity of the channel. Conversely, no scheme can do better. One great benefit of the separation theorem is that a communications link can be used for a large variety of sources: one good channel coding solution can be used with any source. Virtually all systems in use today are based on this principle. It is important though to be aware of the limitations of the source-channel separation theorem. The optimality is only in terms of the achievable distortion when large blocks of data are encoded together. Joint schemes can be substantially better in terms of complexity

or delay. Also, the separation is no longer valid if one looks at multi-user scenarios.

We will not be concerned with the source coding problem or, equivalently, we assume that the source coding problem has been solved. For us, the source emits a sequence of independent identically distributed (iid) bits which are equally likely to be zero or one. Under this assumption, we will see how to accomplish the channel coding problem in an efficient manner for a variety of scenarios.

§1.3. Coding: Trial and Error

How can we transmit information reliably over a noisy channel at a strictly positive rate? At some level we have already given the answer: add redundancy to the message that can be exploited to combat the distortion introduced by the channel. By starting with a special case we want to clarify the key concepts.

Example 1.3 (Binary Symmetric Channel). Consider the *binary symmetric channel* with *cross-over probability* ϵ depicted in Figure 1.4. We denote it by BSC(ϵ). Both

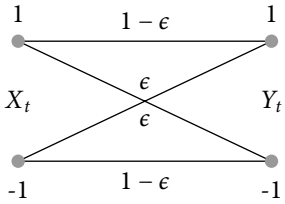

Figure 1.4: BSC(ϵ).

input X_t and output Y_t are elements of $\{\pm 1\}$. A transmitted bit is either received correctly or received *flipped*, the latter occurring with probability ϵ, and different bits are flipped or not flipped independently. We can assume that $0 < \epsilon < \frac{1}{2}$ without loss of generality.

The BSC is the generic model of a binary-input memoryless channel in which hard decisions are made at the front end of the receiver, i.e., where the received value is quantized to two values. ◇

First Trial: Suppose that the transmitted bits are independent and that $\mathbb{P}\{X_t = +1\} = \mathbb{P}\{X_t = -1\} = \frac{1}{2}$. We start by considering *uncoded* transmission over the BSC(ϵ). Thus, we send the source bits across the channel as is, without the insertion of redundant bits. At the receiver we estimate the transmitted bit X based on the observation Y. As we will learn in Section 1.5, the decision rule that minimizes the bit-error probability, call it $\hat{x}^{\text{MAP}}(y)$, is to choose that element of $\{\pm 1\}$ which maximizes $p_{X|Y}(x \,|\, y)$ for the given y. Since the prior on X is uniform, an application of Bayes's rule shows that this is equivalent to maximizing $p_{Y|X}(y \,|\, x)$ for the given

y. Since $\epsilon < \frac{1}{2}$ we conclude that the optimal estimator is $\hat{x}^{\text{MAP}}(y) = y$. The probability that the estimate differs from the true value, i.e., $\text{P}_\text{b} = \mathbb{P}\{\hat{x}^{\text{MAP}}(Y) \neq X\}$, is equal to ϵ. Since for every information bit we want to convey we send exactly one bit over the channel we say that this scheme has *rate* 1. We conclude that with uncoded transmission we can achieve a $(\text{rate}, \text{P}_\text{b})$-pair of $(1, \epsilon)$.

Second Trial: If the error probability ϵ is too high for our application, what transmission strategy can we use to lower it? The simplest strategy is *repetition coding*. Assume we repeat each bit k times. To keep things simple, assume that k is odd. So if X, the bit to be transmitted, has value x then the input to the BSC(ϵ) is the k-tuple $x, \ldots, x$. Denote the k associated observations by $Y_1, \ldots, Y_k$. It is intuitive, and not hard to prove, that the estimator that minimizes the bit-error probability is given by the majority rule

$$\hat{x}^{\text{MAP}}(y_1, \ldots, y_k) = \text{majority of } \{y_1, \ldots, y_k\}.$$

Hence the probability of bit error is given by

$$\text{P}_\text{b} = \mathbb{P}\left\{\hat{x}^{\text{MAP}}(Y) \neq X\right\} \overset{k \text{ odd}}{=} \mathbb{P}\left\{\text{at least } \left\lceil k/2 \right\rceil \text{ errors occur}\right\} = \sum_{i > k/2} \binom{k}{i} \epsilon^i (1-\epsilon)^{k-i}.$$

Since for every information bit we want to convey we send k bits over the channel we say that such a scheme has rate $\frac{1}{k}$. So with repetition codes we can achieve the $(\text{rate}, \text{P}_\text{b})$-pairs $\left(\frac{1}{k}, \sum_{i>k/2} \binom{k}{i} \epsilon^i (1-\epsilon)^{k-i}\right)$. For P_b to approach zero we have to choose k larger and larger and as a consequence the rate approaches zero as well.

Can we keep the rate positive and make the error probability go to zero?

§1.4. Codes and Ensembles

Information is inherently discrete. It is natural and convenient to use *finite* fields to represent it. The most important instance for us is the *binary field* $\mathbb{F}_2$, consisting of $\{0, 1\}$ with mod-2 addition and mod-2 multiplication ($0 + 0 = 1 + 1 = 0$; $0 + 1 = 1$; $0 \cdot 0 = 1 \cdot 0 = 0$; $1 \cdot 1 = 1$). In words, if we use $\mathbb{F}_2$ then we represent information in terms of (sequences of) *bits*, a natural representation and convenient for the purpose of processing. If you are not familiar with finite fields, very little is lost if you replace any mention of a generic finite field $\mathbb{F}$ with $\mathbb{F}_2$. We write $|\mathbb{F}|$ to indicate the number of elements of the finite field $\mathbb{F}$, e.g., $|\mathbb{F}_2| = 2$. Why do we choose finite *fields*? As we will see, by using algebraic operations in both the encoding as well as the decoding we can significantly reduce the complexity.

Definition 1.5 (Code). A *code* C of *length* n and *cardinality* M over a field $\mathbb{F}$ is a collection of M elements from $\mathbb{F}^n$, i.e.,

$$C(n, M) = \{x^{[1]}, \ldots, x^{[M]}\},\ x^{[m]} \in \mathbb{F}^n, 1 \leq m \leq M.$$

The elements of the code are called *codewords*. The parameter n is called the *block-length*. ▽

EXAMPLE 1.6 (REPETITION CODE). Let $\mathbb{F} = \mathbb{F}_2$. The binary *repetition code* of length 3 is defined as $C(n = 3, M = 2) = \{000, 111\}$. ◇

In the preceding example we have introduced *binary* codes, i.e., codes whose components are elements of $\mathbb{F}_2 = \{0, 1\}$. Sometimes it is more convenient to think of the two field elements as $\{\pm 1\}$ instead (see, e.g., the definition of the BSC in Example 1.3). The standard mapping is $0 \leftrightarrow 1$ and $1 \leftrightarrow -1$. It is convenient to use both notations. We freely and frequently switch. With some abuse of notation, we make no distinction between these two cases and talk about binary codes and $\mathbb{F}_2$ even if the components take values in $\{\pm 1\}$.

DEFINITION 1.7 (RATE). The *rate* of a code $C(n, M)$ is $r = \frac{1}{n} \log_{|\mathbb{F}|} M$. It is measured in *information symbols per transmitted symbol.* ▽

EXAMPLE 1.8 (REPETITION CODE). Let $\mathbb{F} = \mathbb{F}_2$. We have $r(C(3, 2)) = \frac{1}{3} \log_2 2 = \frac{1}{3}$. It takes three channel symbols to transmit one information symbol. ◇

The following two definitions play a role only much later in the book, but it is convenient to collect them here for reference.

DEFINITION 1.9 (SUPPORT SET). The *support set* of a codeword $x \in C$ is the set of locations $i \in [n] = \{1, \ldots, n\}$ such that $x_i \neq 0$. ▽

DEFINITION 1.10 (MINIMAL CODEWORDS). Consider a *binary* code C, i.e., a code over $\mathbb{F}_2$. We say that a codeword $x \in C$ is *minimal* if its support set does not contain the support set of any other (non-zero) codeword. ▽

The Hamming distance introduced in the following definition and the derived minimum distance of a code (see Definition 1.12) are *the* central characters in all of classical coding. For us they only play a minor role. This is probably one of the most distinguishing factors between classical and modern coding.

DEFINITION 1.11 (HAMMING WEIGHT AND HAMMING DISTANCE). Let $u, v \in \mathbb{F}^n$. The *Hamming weight* of a word u, which we denote by $w(u)$, is equal to the number of non-zero symbols in u, i.e., the cardinality of the support set. The *Hamming distance* of a pair (u, v), which we denote by $d(u, v)$, is the number of positions in which u differs from v. We have $d(u, v) = d(u - v, 0) = w(u - v)$. Further, $d(u, v) = d(v, u)$ and $d(u, v) \geq 0$, with equality if and only if $u = v$. Also, $d(\cdot, \cdot)$ satisfies the *triangle inequality*

$$d(u, v) \leq d(u, t) + d(t, v),$$

for any triple $u, v, t \in \mathbb{F}^n$. In words, $d(\cdot, \cdot)$ is a true *distance* in the mathematical sense (see Problem 1.2). ▽

DEFINITION 1.12 (MINIMUM DISTANCE OF A CODE). Let C be a code. Its *minimum distance* $d(C)$ is defined as

$$d(C) = \min\{d(u,v) : u, v \in C, u \neq v\}. \qquad \nabla$$

Let $x \in \mathbb{F}^n$ and $t \in \mathbb{N}$. A *sphere* of radius t centered at the point x is the set of all points in $\mathbb{F}^n$ that have distance at most t from x. If, for a code C of minimum distance d, we place spheres of radius $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ around each codeword, then these spheres are disjoint. This follows from the triangle inequality: if u and v are codewords and x is any element in $\mathbb{F}^n$ then $d(C) \le d(u,v) \le d(u,x) + d(v,x)$. If x is in the sphere of radius t around u then this implies that $d(v,x) \ge d(C) - d(u,x) \ge \frac{d+1}{2} > t$. In words, x is not in the sphere of radius t around v. Further, by definition of d, t is the largest such radius.

The radius t has an important operational meaning that explains why much of classical coding is centered on the construction of codes with large minimum distance. To be concrete, consider the binary case. Assume we use a code $C(n, M, d)$ (i.e., a code with M codewords of length n and minimum distance d) for transmission over a BSC and assume that we employ a *bounded distance* decoder with decoding radius t, $t \le \left\lfloor \frac{d-1}{2} \right\rfloor$. More precisely, given y the decoder chooses $\hat{x}^{\mathrm{BD}}(y)$ defined by

$$\hat{x}^{\mathrm{BD}}(y) = \begin{cases} x \in C, & \text{if } d(x,y) \le t, \\ \text{error}, & \text{if no such } x \text{ exists}, \end{cases}$$

where by "error" the decoder declares that it is unable to decode. As we have just discussed, there can be *at most one* $x \in C$ so that $d(x,y) \le t$. Therefore, if the weight of the error does not exceed t, then such a combination finds the correct transmitted word. A large t hence implies a large resilience against channel errors.

How large can d (and hence t) be made in the binary case? Let $\delta = d/n$ denote the *normalized* distance and consider for a fixed rate r, $0 < r < 1$,

$$\delta^*(r) = \limsup_{n\to\infty} \max\left\{ \frac{d(C)}{n} : C \in \mathcal{C}\left(n, 2^{\lfloor nr \rfloor}\right) \right\},$$

where $\mathcal{C}\left(n, 2^{\lfloor nr \rfloor}\right)$ denotes the set of all binary block codes of length n containing at least $2^{\lfloor nr \rfloor}$ codewords. Problem 1.15 discusses the asymptotic *Gilbert-Varshamov* bound

$$h_2^{-1}(1-r) \le \delta^*(r),$$

where $h_2(x) = -x\log_2 x - (1-x)\log_2(1-x)$ is the *binary entropy function* and where for $y \in [0,1]$, $h_2^{-1}(y)$ is the unique element $x \in [0, \frac{1}{2}]$ such that $h_2(x) = y$. *Elias* introduced the following upper bound,

$$\delta^*(r) \le 2h_2^{-1}(1-r)\left(1 - h_2^{-1}(1-r)\right). \tag{1.13}$$

Both bounds are illustrated in Figure 1.14. We can now answer the question posed

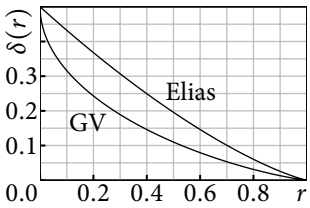

Figure 1.14: Upper and lower bound on $\delta^*(r)$.

at the end of the previous section. For a fixed channel BSC(ϵ) pick a rate r such that $\delta^*(r) > 2\epsilon + \omega$, where ω is some arbitrarily small but strictly positive quantity. We see from the Gilbert-Varshamov bound that such a strictly positive r and ω exist if $\epsilon < 1/4$. By the definition of δ^*, we can find a code of rate r of arbitrarily large blocklength n which has a relative minimum distance at least $\delta = 2\epsilon + \omega$. By Chebyshev's inequality (see Lemma C.3 on page 480), for every positive probability P bounded away from 1 there exists a positive constant c such that the number of channel flips in a block of length n is at most $n\epsilon + c\sqrt{n}$ with probability P. Assume that we employ a bounded distance decoder. If we choose n sufficiently large so that $n\epsilon + c\sqrt{n} < \delta n/2 = n\epsilon + n\omega/2$, then the bounded distance decoder succeeds with probability at least P. Since P can be chosen arbitrarily close to 1 we see that there exist codes that allow transmission at a positive rate with arbitrarily small positive probability of error. The above procedure is by no means optimal and does not allow us to determine up to what rates reliable transmission is possible. We will see in Section 1.6 how we can characterize the *largest* such rate.

Constructing provably good codes is difficult. A standard approach to show the *existence* of good codes is the probabilistic method: an *ensemble* $\mathcal{C}$ of codes is "constructed" using some random process and one proves that good codes occur with positive probability within this ensemble. Often the probability is close to 1 – almost all codes are good. This approach, used already by Shannon in his 1948 landmark paper, simplifies the code "construction" task enormously (at the cost of a less useful result).

Definition 1.15 (Shannon's Random Ensemble). Let the field $\mathbb{F}$ be fixed. Consider the following ensemble $\mathcal{C}(n, M)$ of codes of length n and cardinality M. There are nM degrees of freedom in choosing a code, one degree of freedom for each component of each codeword. The ensemble consists of all $|\mathbb{F}|^{nM}$ possible codes of length n and cardinality M. We endow this set with a uniform probability distribution. To sample from this ensemble proceed as follows. Pick the codewords

$x^{[1]}, \ldots, x^{[M]}$ randomly by letting each component $x_i^{[m]}$ be an independently and uniformly chosen element of $\mathbb{F}$. ▽

We will see that such a code is likely to be "good" for many channels.

§1.5. MAP and ML Decoding and APP Processing

Assume we transmit over a channel with input $\mathbb{F}$ and output space $\mathcal{Y}$ using a code $C(n, M) = \{x^{[1]}, \ldots, x^{[M]}\}$. Let the channel be specified by its transition probability $p_{Y|X}(y|x)$. The transmitter chooses the codeword $X \in C(n, M)$ with probability $p_X(x)$. (In communications the idea is that the transmitter wants to transmit one of M messages and uses one codeword for each possible message.) This codeword is then transmitted over the channel. Let Y denote the observation at the output of the channel. To what codeword should Y be decoded? If we decode Y to $\hat{x}(Y) \in C$, then the probability that we have made an error is $1 - p_{X|Y}(\hat{x}(Y)|y)$. Thus, to minimize the probability of block error we should choose $\hat{x}(Y)$ to maximize $p_{X|Y}(\hat{x}(Y)|y)$. The *maximum a posteriori* (MAP) decoding rule reads

$$\begin{aligned} \hat{x}^{\mathrm{MAP}}(y) &= \operatorname{argmax}_{x \in C} p_{X|Y}(x|y) \\ \text{by Bayes's rule} \qquad &= \operatorname{argmax}_{x \in C} p_{Y|X}(y|x) \frac{p_X(x)}{p_Y(y)} \\ &= \operatorname{argmax}_{x \in C} p_{Y|X}(y|x) p_X(x). \end{aligned}$$

Ties can be broken in some arbitrary manner without affecting the error probability. As we indicated, this estimator minimizes the probability of (block) error $\mathrm{P_B} = \mathbb{P}\{\hat{x}^{\mathrm{MAP}}(Y) \neq X\}$. If all codewords are equally likely, i.e., if p_X is uniform, then

$$\hat{x}^{\mathrm{MAP}}(y) = \operatorname{argmax}_{x \in C} p_{Y|X}(y|x) p_X(x) = \operatorname{argmax}_{x \in C} p_{Y|X}(y|x) = \hat{x}^{\mathrm{ML}}(y),$$

where the right-hand side represents the decoding rule of the *maximum likelihood* (ML) decoder. In words, for a uniform prior p_X the MAP and the ML decoders are equivalent.

The key step in the MAP decoding process is to compute the *a posteriori probability* (APP) $p_{X|Y}(x|y)$, i.e., the distribution of X given the observation Y. So we call a MAP decoder also an APP decoder. Also, we will say that we perform *APP processing* to mean that we compute the a posteriori probabilities.

§1.6. Channel Coding Theorem

We have already seen that transmission at a strictly positive rate and an arbitrarily small positive probability of error is possible. What is the *largest* rate at which we

can achieve a vanishing probability of error? Let us now investigate this question for transmission over the BSC.

We are interested in the scenario depicted in Figure 1.16. For a given binary code $C(n, M)$ the transmitter chooses with uniform probability a codeword $X \in C(n, M)$ and transmits this codeword over the channel BSC(ϵ). The output of the channel is denoted by Y. At the receiver the decoder estimates the transmitted codeword given the observation Y using the MAP rule $\hat{x}^{\text{MAP}}(y)$. How small can we make the incurred *block error probability* $\mathrm{P}_{\mathrm{B}}^{\text{MAP}}(C, \epsilon) = \mathbb{P}\{\hat{x}^{\text{MAP}}(Y) \neq X\}$ for given parameters n and M? Let $\hat{\mathrm{P}}_{\mathrm{B}}^{\text{MAP}}(n, M, \epsilon)$ be the minimum of $\mathrm{P}_{\mathrm{B}}^{\text{MAP}}(C, \epsilon)$ over all choices of $C \in \mathcal{C}(n, M)$.

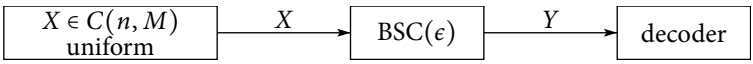

Figure 1.16: Transmission over the BSC(ϵ).

THEOREM 1.17 (SHANNON'S CHANNEL CODING THEOREM). *If $0 < r < 1 - h_2(\epsilon)$ then $\hat{\mathrm{P}}_{\mathrm{B}}^{\text{MAP}}(n, 2^{\lfloor rn \rfloor}, \epsilon) \overset{n\to\infty}{\longrightarrow} 0$.*

Proof. Pick a code C from Shannon's random ensemble $\mathcal{C}(n, 2^{\lfloor rn \rfloor})$ introduced in Definition 1.15. Since the MAP decoder is hard to analyze we use the following suboptimal decoder. For some fixed Δ, $\Delta > 0$, define $\rho = n\epsilon + \sqrt{2n\epsilon(1-\epsilon)/\Delta}$. If $x^{[m]}$ is the only codeword such that $d(y, x^{[m]}) \leq \rho$ then decode y as $x^{[m]}$ – otherwise declare an error.

For $u, v \in \{\pm 1\}^n$ let

$$f(u, v) = \begin{cases} 0, & \text{if } d(u, v) > \rho, \\ 1, & \text{if } d(u, v) \leq \rho, \end{cases}$$

and define

$$g^{[m]}(y) = 1 - f(x^{[m]}, y) + \sum_{m' \neq m} f(x^{[m']}, y).$$

Note that $g^{[m]}(y)$ equals zero if $x^{[m]}$ is the only codeword such that $d(y, x^{[m]}) \leq \rho$ and that it is at least one otherwise. Let $\mathrm{P}_{\mathrm{B}}^{[m]}$ denote the conditional block error probability assuming that $X = x^{[m]}$, i.e., $\mathrm{P}_{\mathrm{B}}^{[m]} = \mathbb{P}\{\hat{x}(Y) \neq X \,|\, X = x^{[m]}\}$. We have

$$\begin{aligned} \mathrm{P}_{\mathrm{B}}^{[m]}(C, \epsilon) &= \sum_{y : g^{[m]}(y) \geq 1} p_{Y|X^{[m]}}(y \,|\, x^{[m]}) \leq \sum_{y \in \{\pm 1\}^n} p_{Y|X^{[m]}}(y \,|\, x^{[m]}) g^{[m]}(y) \\ &= \sum_{y \in \{\pm 1\}^n} p_{Y|X^{[m]}}(y \,|\, x^{[m]})[1 - f(x^{[m]}, y)] \end{aligned}$$