

In this topic

Topic 1 detected one symbol at a time and showed how errors accumulate along a message. We now study how redundancy improves reliability. The transmitter sends a codeword of n symbols for every k information bits, and the receiver decides on the whole codeword. Sections 2.1 and 2.2 correspond to the first lecture and Sections 2.3 and 2.4 to the second.

- We review codes, information bits, coded bits and rate, using repetition on the bit-flipping channel.
- We extend MAP and ML detection to codewords. We obtain distance rules for Gaussian and bit-flipping channels, consistency for erasures, and a one-sided rule for exponential noise.
- We compare hard and soft decisions for the same code on the Gaussian channel, and identify the information lost by deciding symbolwise.
- We compare codes at fixed symbol energy and at fixed energy per information bit. Repetition improves reliability at vanishing rate, motivating the positive-rate question of Topic 3.

2.1 Codes and rate

In Topic 1 the transmitter sent one symbol for every $\log_2 M$ bits, and the receiver decided symbolwise. Exercise 1.6 of Topic 1 showed the weakness of this scheme: the probability that a sequence of n symbols is entirely correct is $(1 - P_e)^n$, which vanishes as n grows however small $P_e > 0$ is. To send long messages reliably, we need something better than sending every bit once. The remedy is **redundancy**, that is, to use more symbols than strictly necessary, so that the receiver can recover from some of the errors.

Definition 2.1. An **encoder** maps every sequence of k information bits $(D_1, \dots, D_k)$ to a **codeword** $\mathbf{x} = (x_1, \dots, x_n)$ of n symbols of the constellation $\mathcal{X}$. Different information sequences have different codewords. The set $\mathcal{C}$ of the 2^k codewords is the **code**, and its **rate**, in bits per channel use, is

$$R = \frac{k}{n}. \quad (2.1)$$

Figure 2.1 shows the position of the encoder and the decoder. The uncoded transmission of Topic 1 is the special case in which every symbol carries $\log_2 M$ bits, with $M = |\mathcal{X}|$ a power of two, that is, $R = \log_2 M$. Coding means choosing R smaller than that, and the whole question of this topic and the next is what we obtain in exchange.



Figure 2.1: Coded transmission: k information bits become a codeword $\mathbf{x} = (x_1, \dots, x_n)$, and the decoder sees $\mathbf{y} = (y_1, \dots, y_n)$.

The simplest code is the **repetition code**, which has $k = 1$ and sends the single information bit D a total of n times. With s_0 and s_1 the two points of a binary constellation, the two codewords, indexed by $D \in \{0, 1\}$, are

$$\mathbf{x}(D) = (s_D, \dots, s_D), \quad (2.2)$$

and the rate is $R = 1/n$. For binary constellations, an encoder first produces n **coded bits** $c_1, \dots, c_n$ from the k information bits, and then maps each coded bit to a symbol, $x_i = s_{c_i}$. For the repetition code, $c_1 = \dots = c_n = D$. On the bit-flipping and bit-erasing channels the constellation is $\{0, 1\}$. On the Gaussian channel we take $s_0 = -A$ and $s_1 = A$, and on the exponential channel $s_0 = 0$ and $s_1 = 2A$, with $A > 0$. As in Topic 1, Gaussian inputs are amplitudes, whereas exponential-channel inputs are themselves energies.

When the constellation has $M > 2$ points, the encoder produces n_c coded bits and maps $\log_2 M$ of them to each symbol. Its binary rate is $r = k/n_c$, and the rate in bits per channel use is $R = r \log_2 M$. In this topic all the symbols are binary, so $n_c = n$ and $R = r$.

Let us recall how the repetition code behaves on the bit-flipping channel with flip probability $p < \frac{1}{2}$, which was studied in the information theory course. The natural decoder is **majority voting**: decide the bit that appears most often among the n received bits, and toss a fair coin in case of a tie. For $n = 3$, Table 2.1 lists the eight possible outputs when $D = 0$ is sent, together with their probabilities and the decision of the majority decoder.

(y_1, y_2, y_3)	$W(\mathbf{y} 000) = \prod_{i=1}^3 W(y_i 0)$	Decision $\hat{D}$	Error
000	$(1-p)^3$	0	no
001, 010, 100	$p(1-p)^2$ each	0	no
011, 101, 110	$p^2(1-p)$ each	1	yes
111	p^3	1	yes

Table 2.1: The repetition code with $n = 3$ on the bit-flipping channel, when $D = 0$ is sent.

The table is once more an application of Table 1.1 of Topic 1. Every probability in the second column is a product, because the channel is memoryless, and the conditional error probability is the sum of the rows that lead to an error,

$$P_e(0) = 3p^2(1-p) + p^3 = p^2(3-2p). \quad (2.3)$$

By symmetry $P_e(1)$ is the same, so $P_e = p^2(3-2p)$ whatever the prior, and this is smaller than p exactly when $0 < p < \frac{1}{2}$, since $p - p^2(3-2p) = p(1-p)(1-2p)$.

In general, for odd n , the majority decoder fails when at least $(n+1)/2$ of the n bits are flipped, so

$$P_e = \sum_{i=(n+1)/2}^n \binom{n}{i} p^i (1-p)^{n-i}. \quad (2.4)$$

For $n = 3$ the sum has the two terms of (2.3). The case of even n , with its ties, is Exercise 2.2. For $p < \frac{1}{2}$ this probability goes to zero as n grows, because the fraction of flipped bits concentrates around p , which is below one half.

Hence, the repetition code makes the error probability as small as we wish. The cost is the rate, since $R = 1/n$ goes to zero at the same time. We will come back to this trade-off in Section 2.4.

2.2 Optimal decoding

We now place the decoder in the framework of Topic 1. The transmitter has a message to send, which we may think of as the k bits or, more conveniently, as an index m in a set $\mathcal{M}$ of 2^k values with prior probabilities $\Pr(m)$. For the repetition code, m is simply the bit D . The message selects the codeword $\mathbf{x}(m)$, the channel produces $\mathbf{y} = (y_1, \dots, y_n)$, a realization of the random sequence $\mathbf{Y}$, and a **decoder** is any function g from the sequences $\mathbf{y}$ to the messages, $\hat{m} = g(\mathbf{y})$. Its error probabilities are defined as in Topic 1. The conditional error probability when m is sent is

$$P_e(m) = \Pr(g(\mathbf{Y}) \neq m \mid m \text{ sent}), \quad (2.5)$$

and the average error probability is

$$P_e = \sum_{m \in \mathcal{M}} \Pr(m) P_e(m). \quad (2.6)$$

Theorem 2.2. No decoder has a smaller average error probability than the MAP decoder,

$$g_{\text{MAP}}(\mathbf{y}) = \arg \max_{m \in \mathcal{M}} \Pr(m) W(\mathbf{y}|\mathbf{x}(m)). \quad (2.7)$$

With equiprobable messages, the MAP decoder is the ML decoder,

$$g_{\text{ML}}(\mathbf{y}) = \arg \max_{m \in \mathcal{M}} W(\mathbf{y}|\mathbf{x}(m)). \quad (2.8)$$

Proof. Nothing in the proof of Theorem 1.4 of Topic 1 used that the observation was a single number. Replace the symbol x by the message m , the output y by the sequence $\mathbf{y}$, and the channel law $W(y|x)$ by $W(\mathbf{y}|\mathbf{x}(m))$, and the proof remains valid without changes. $\square$

A tie between maximizers may be broken in any way, and we break it with a fair coin. Note that P_e is the probability of a wrong message, that is, of a wrong codeword. For the repetition code the message is a single bit, so it is also the bit error probability.

The new element is the channel law of a sequence. As in Topic 1, W denotes a probability for discrete outputs and a density for continuous outputs. All our channels are memoryless, so by the last line of Table 1.1 of Topic 1,

$$W(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^n W(y_i|x_i), \quad (2.9)$$

and the ML decoder maximizes a product of n factors. Since the logarithm is increasing, with the convention $\ln 0 = -\infty$, we may equally maximize the sum of the logarithms,

$$g_{\text{ML}}(\mathbf{y}) = \arg \max_{m \in \mathcal{M}} \sum_{i=1}^n \ln W(y_i|x_i(m)). \quad (2.10)$$

In words, every channel output y_i gives a score to each candidate codeword, namely the logarithm of how likely y_i is under that codeword, and the decoder adds the scores and picks the codeword with the highest total. This sum is called the **log-likelihood** of the codeword. Table 2.2 lists the scores on the channels of the course. The terms that do not depend on the codeword have been dropped, since they do not affect the comparison. For the repetition code, for instance, every $x_i(m)$ is the same symbol s_m , so the decoder compares just two numbers, $\sum_{i=1}^n \ln W(y_i|s_0)$ and $\sum_{i=1}^n \ln W(y_i|s_1)$. For MAP decoding, we add $\ln \Pr(m)$ to the log-likelihood of each codeword. Let us see what the sums of these scores become on the channels of the course.

Channel	Score of y_i for a codeword with x_i in position i
Real Gaussian	$-(y_i - x_i)^2 / (2\sigma^2)$
Complex Gaussian	$- y_i - x_i ^2 / \sigma^2$
Bit-flipping	$\ln(1-p)$ if $y_i = x_i$, $\ln p$ otherwise
Bit-erasing	$\ln(1-p)$ if $y_i = x_i$, $\ln p$ if $y_i = ?$, $-\infty$ if $y_i \in \{0,1\}$ and $y_i \neq x_i$
Exponential noise	$-(y_i - x_i)/\mu$ if $y_i \geq x_i$, $-\infty$ otherwise

Table 2.2: The symbolwise scores $\ln W(y_i|x_i)$ of (2.10), up to terms that do not depend on the codeword.

Proposition 2.3. On the channels of Table 1.2 of Topic 1, the ML decoder takes the following forms.

- (a) Real and complex Gaussian: the codeword nearest to $\mathbf{y}$ in Euclidean distance,

$$g_{\text{ML}}(\mathbf{y}) = \arg \min_{m \in \mathcal{M}} \|\mathbf{y} - \mathbf{x}(m)\|^2, \quad (2.11)$$

where $\|\mathbf{y} - \mathbf{x}\|^2 = \sum_{i=1}^n |y_i - x_i|^2$.

- (b) Bit-flipping with $p < \frac{1}{2}$: the codeword nearest to $\mathbf{y}$ in Hamming distance,

$$g_{\text{ML}}(\mathbf{y}) = \arg \min_{m \in \mathcal{M}} d_{\text{H}}(\mathbf{y}, \mathbf{x}(m)), \quad (2.12)$$

where d_{H} counts the positions in which two sequences differ.

- (c) Bit-erasing: any codeword that agrees with $\mathbf{y}$ in all the positions that were not erased. If there are several, they are equally likely and any of them may be chosen. The MAP decoder breaks the tie with the priors.
(d) Exponential noise: among the codewords with $x_i(m) \leq y_i$ in every position, the one with the largest total energy $\sum_{i=1}^n x_i(m)$.

Proof. In each case we write the product (2.9) and drop the factors that do not depend on m . Recall that a product of exponentials is the exponential of the sum, which is the reason we took logarithms in (2.10). (a) On the complex channel, the product of the Gaussian densities is $(\pi\sigma^2)^{-n} \exp(-\|\mathbf{y} - \mathbf{x}\|^2/\sigma^2)$, decreasing in the distance. On the real channel the exponent has $2\sigma^2$ and the factor in front is $(2\pi\sigma^2)^{-n/2}$, and the conclusion is the same. (b) If $\mathbf{y}$ and $\mathbf{x}$ differ in d positions, the product is $p^d(1-p)^{n-d} = (1-p)^n(p/(1-p))^d$, and $p/(1-p) < 1$, so the product decreases with d . (c) If $\mathbf{y}$ has e erasures, the product is $p^e(1-p)^{n-e}$ for every codeword that agrees with the unerased positions, and 0 for the others. (d) The product is $\mu^{-n} \exp(-\sum_{i=1}^n (y_i - x_i)/\mu)$ if $y_i \geq x_i$ for all i , and 0 otherwise. For fixed $\mathbf{y}$, it grows with $\sum_{i=1}^n x_i$. $\square$

Example 2.4. For the repetition code, the four rules take a simple form.

- (a) Real Gaussian, $\mathcal{X} = \{-A, +A\}$: expanding the squares, $\|\mathbf{y} - (A, \dots, A)\|^2 < \|\mathbf{y} + (A, \dots, A)\|^2$ is the same as $\sum_{i=1}^n y_i > 0$. The ML decoder adds the n observations and looks at the sign of the sum. On the complex channel, it adds the real parts.
- (b) Bit-flipping: the Hamming distance to $(0, \dots, 0)$ is the number of ones in $\mathbf{y}$, and the distance to $(1, \dots, 1)$ the number of zeros. The ML decoder is the majority vote of Section 2.1, which is therefore optimal for equiprobable bits and $p < \frac{1}{2}$.
- (c) Bit-erasing: the two codewords are constant sequences, so a single unerased bit decides. If all n bits are erased, the decoder tosses a coin.
- (d) Exponential noise, $\mathcal{X} = \{0, 2A\}$: the codeword $(2A, \dots, 2A)$ is possible only if every $y_i \geq 2A$, and then it has the larger energy. Hence, the decoder decides 1 if all the observations are at least $2A$, and 0 otherwise.

Problem 2.5. Compute the error probability of the repetition code with the ML decoder on the real Gaussian channel, as a function of n and of the SNR A^2/σ^2 of one symbol.

Solution: Suppose $D = 0$ is sent, that is, $\mathbf{x} = (-A, \dots, -A)$. Write $Y_i = -A + Z_i$, where the Z_i are independent zero-mean Gaussian noise samples of variance σ^2 . The sum of the observations is

$$\sum_{i=1}^n Y_i = -nA + \sum_{i=1}^n Z_i, \quad (2.13)$$

and the decoder errs when this sum is positive, that is, when $\sum_{i=1}^n Z_i > nA$. The sum of n independent Gaussian variables of variance σ^2 is Gaussian with variance $n\sigma^2$. The variances add because the variables are independent, and the sum of independent Gaussians is again Gaussian, as seen in the probability course. Hence, by equation (1.23) of Topic 1 with threshold nA and variance $n\sigma^2$,

$$P_e(0) = \Pr\left(\sum_{i=1}^n Z_i > nA\right) = \frac{1}{2} \operatorname{erfc}\left(\frac{nA}{\sqrt{2n}\sigma}\right) = \frac{1}{2} \operatorname{erfc}\left(\sqrt{n \operatorname{SNR}/2}\right). \quad (2.14)$$

By symmetry $P_e(1)$ is the same, so for any prior

$$P_e = \frac{1}{2} \operatorname{erfc}\left(\sqrt{n \operatorname{SNR}/2}\right). \quad (2.15)$$

The function erfc decreases with its argument, so collecting more observations reduces the error probability. Compared with the uncoded 2-PAM of equation (1.22) of Topic 1, the SNR is multiplied by n , because the decoder collects the energy of the n copies. For $n = 2$ this is the same as doubling the energy of a single symbol, a gain of 3 dB.

Problem 2.6. Compute the error probability of the repetition code with equiprobable bits and the ML decoder on the bit-erasing and exponential noise channels.

Solution: On the bit-erasing channel, the decoder of Example 2.4(c) errs only if all n bits are erased, which happens with probability p^n by the product rule, and the fair coin then gives the wrong bit, with probability $\frac{1}{2}$. Hence, $P_e = \frac{1}{2}p^n$. Note the two different quantities: p^n is the probability that everything is erased, and $\frac{1}{2}p^n$ is the probability of an error. With unequal priors, the MAP decoder guesses the more probable bit at a total erasure and errs with probability $p^n \min(\Pr(0), \Pr(1))$, as in Exercise 1.1 of Topic 1.

On the exponential channel, when $D = 1$ is sent every y_i is at least $2A$ and the decision is always right, $P_e(1) = 0$. When $D = 0$ is sent, an error requires every noise sample to reach $2A$. The samples are independent, so by the product rule and Example 1.14 of Topic 1,

$$P_e(0) = \prod_{i=1}^n \Pr(Z_i \geq 2A) = e^{-2nA/\mu}, \quad (2.16)$$

and with equiprobable bits $P_e = \frac{1}{2}e^{-2nA/\mu}$. Since the bits are equiprobable, $E_s = A$. In terms of the SNR $E_s/\mu = A/\mu$ of one symbol, $P_e = \frac{1}{2}e^{-2n \operatorname{SNR}}$. Once more, n repetitions are worth n times the SNR.

Problem 2.7. The repetition code with $n = 3$ is used on the bit-flipping channel with $p = \frac{1}{10}$, and the bit to be sent is 1 with probability $\frac{19}{20}$. What does the MAP decoder decide for each received sequence, and what is its error probability?

Solution: Let d be the number of ones in $\mathbf{y}$. By (2.9), $W(\mathbf{y}|000) = p^d(1-p)^{3-d}$ and $W(\mathbf{y}|111) = p^{3-d}(1-p)^d$, so the two joint probabilities that the MAP decoder compares depend on $\mathbf{y}$ only through d . Table 2.3 lists them.

d	20000 $\Pr(0)W(\mathbf{y} 000)$	20000 $\Pr(1)W(\mathbf{y} 111)$	Decision $\hat{D}$
0	729	19	0
1	81	171	1
2	9	1539	1
3	1	13851	1

Table 2.3: The two joint probabilities compared by the MAP decoder of Problem 2.7, multiplied by the common denominator 20000, for a sequence with d ones.

The decoder decides 1 as soon as a single 1 is received, against the majority. In general, the MAP decoder decides 1 when $19p^{3-d}(1-p)^d > p^d(1-p)^{3-d}$. Collecting the powers of $1-p$ on the left and those of p on the right, this is $19(1-p)^{2d-3} > p^{2d-3}$, that is,

$$19 > \left(\frac{1-p}{p}\right)^{3-2d} = 9^{3-2d}. \quad (2.17)$$

For $d = 0$ the right-hand side is 729 and the decoder decides 0. For $d = 1$ it is 9, smaller than 19, and the decoder decides 1. For $d = 2$ and $d = 3$ it is smaller still, in agreement with the table. With such a prior, a single 1 among three bits is more likely a correct bit from a transmitted 1 than a flipped bit from a transmitted 0.

The decoder errs when 0 is sent and some bit is flipped, or when 1 is sent and all three are, so

$$P_e = \frac{1}{20} \left(1 - \left(\frac{9}{10}\right)^3\right) + \frac{19}{20} \left(\frac{1}{10}\right)^3 = \frac{29}{2000}, \quad (2.18)$$

against $p^2(3-2p) = \frac{7}{250} = \frac{56}{2000}$ for majority voting, whose error probability does not depend on the prior. As in Problems 1.7 and 1.9 of Topic 1, the prior can change the decision and reduce the average error probability.

2.3 Hard and soft decisions

We now compare two receivers for the same repetition code on the real Gaussian channel, with equiprobable bits and $\text{SNR} = A^2/\sigma^2$. The ML decoder of Example 2.4(a) works with the observations y_i themselves. A different receiver, natural after Topic 1, first detects every symbol with the symbolwise detector of Topic 1, obtaining a bit $\hat{c}_i$ per symbol, and then decodes from the n detected bits. Figure 2.2 shows the two receivers. The upper one is said to take **hard decisions**, since every observation is reduced to a bit before decoding. The lower one takes **soft decisions**, because the decoder sees the observations in full.

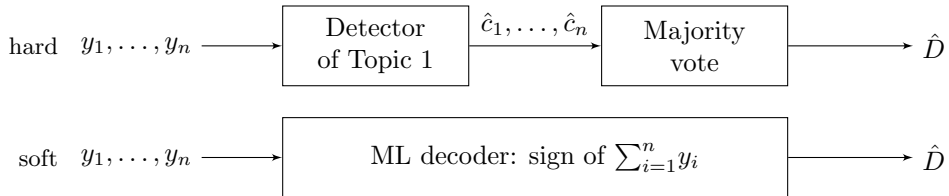


Figure 2.2: Two receivers for the repetition code on the Gaussian channel.

The hard receiver is easy to analyze with the tools we have. Each detected bit $\hat{c}_i$ is wrong with the probability (1.22) of Topic 1, which we call p here,

$$p = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\text{SNR}/2}\right), \quad (2.19)$$

and the errors in different positions are independent, since the noise samples are. In other words, seen from the coded bits to the detected bits, the Gaussian channel followed by the detector is a bit-flipping channel with flip

probability p . On that channel, majority voting is the ML decoder, and for odd n its error probability is (2.4) with the value (2.19) of p . Even lengths include the tie term of Exercise 2.2. The soft receiver has the error probability (2.15).

Example 2.8. Take $n = 2$. The hard receiver sees two detected bits. If they agree, it follows them, and if they disagree, it tosses a coin. An error occurs when both bits are wrong, with probability p^2 , or when they disagree and the coin is wrong, with probability $2p(1 - p) \cdot \frac{1}{2}$. The total is $p^2 + p(1 - p) = p$, so the hard receiver with $n = 2$ is exactly as good as sending the bit once. The soft receiver, on the other hand, has $P_e = \frac{1}{2} \operatorname{erfc}(\sqrt{\operatorname{SNR}})$, the 3 dB gain of Problem 2.5. The second copy of the bit is not wasted, but only if the receiver looks at the two observations together.

Table 2.4 gives the error probability of both receivers for n up to 5, and Figure 2.3 plots it. The soft receiver is better at every SNR, and the difference grows with n . For $n = 3$, for instance, the hard receiver takes the uncoded p down to $p^2(3 - 2p)$, whereas the soft receiver multiplies the argument of erfc by $\sqrt{3}$. At high SNR the loss of the hard receiver tends to $10 \log_{10}(2n/(n + 1))$, from 1.8 dB for $n = 3$ to 3 dB for large n , as Exercise 2.8 shows. Note that the SNR in the table and the figure is that of one symbol. Section 2.4 shows what the comparison looks like at equal energy per bit.

n	Hard decisions	Soft decisions
1	p	$\frac{1}{2} \operatorname{erfc}(\sqrt{\operatorname{SNR}/2})$
2	p	$\frac{1}{2} \operatorname{erfc}(\sqrt{\operatorname{SNR}})$
3	$3p^2(1 - p) + p^3$	$\frac{1}{2} \operatorname{erfc}(\sqrt{3\operatorname{SNR}/2})$
5	$10p^3(1 - p)^2 + 5p^4(1 - p) + p^5$	$\frac{1}{2} \operatorname{erfc}(\sqrt{5\operatorname{SNR}/2})$

Table 2.4: Error probability of the repetition code on the real Gaussian channel with hard and soft decisions, where $p = \frac{1}{2} \operatorname{erfc}(\sqrt{\operatorname{SNR}/2})$ is the error probability of one detected bit. The SNR is that of one symbol.

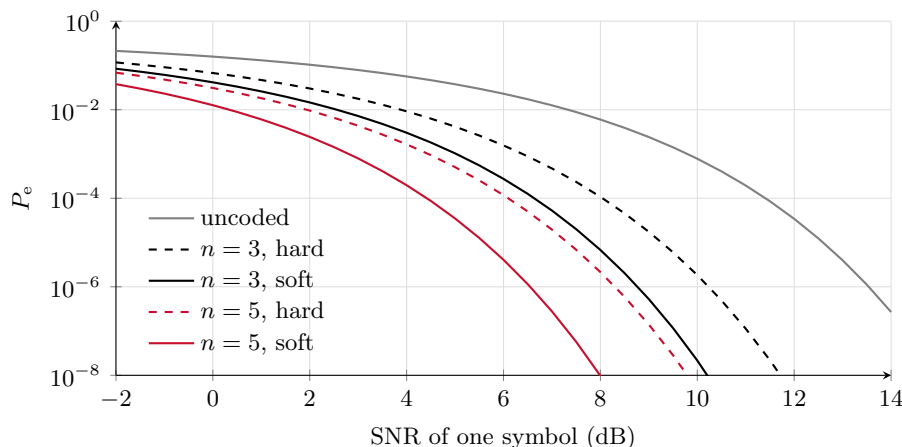


Figure 2.3: The curves of Table 2.4. Dashed lines are hard decisions, solid lines soft decisions.

The loss has a simple origin. The symbolwise detector keeps only the sign of each y_i and discards its magnitude. An observation close to the threshold is nearly useless, and one far from it is nearly certain, but after the detector both count as one vote. The soft decoder adds the observations, so a large y_i can outweigh two small ones of the opposite sign. For instance, with $\mathbf{y} = (2, -\frac{1}{2}, -\frac{1}{2})A$ the hard receiver sees $(+, -, -)$ and decides 0, while the soft receiver computes $\sum_{i=1}^3 y_i = A > 0$ and decides 1.

This is the general lesson of the section. The part of the receiver that processes the channel output should not take decisions. It should pass the observations, or the scores of (2.10), to the decoder. Modern receivers do exactly this, and the scores are then called **soft information**.

The loss of hard decisions depends on the channel. On the exponential channel, the detector of Topic 1 loses nothing at all, because the only information in y_i that the decoder of Example 2.4(d) uses is whether y_i reaches $2A$, and this is exactly what the detector keeps, as Exercise 2.5 shows. In fact, on the exponential channel the ML and MAP decoders of any code depend on each y_i only through the largest constellation point below it, because the rest of the dependence on $\mathbf{y}$ in Proposition 2.3(d) is common to all codewords.

2.4 Rate and reliability

Repetition drives the error probability to zero as n grows: on the bit-flipping channel with fixed $p < \frac{1}{2}$, by (2.4), on the erasure channel with fixed $p < 1$ and on the Gaussian and exponential channels with fixed symbol energy and noise level, by Problem 2.6 and (2.15). The only exceptions are the useless channels, the BSC with $p = \frac{1}{2}$ and the BEC with $p = 1$, where $P_e = \frac{1}{2}$ for every n . The price is the rate, since $R = 1/n$ vanishes too, and on the Gaussian and exponential channels each information bit costs the energy of n symbols. Table 2.5 shows the price on the bit-flipping channel with $p = \frac{1}{10}$. To send k bits reliably we need nk channel uses, and n has to grow without bound. The table gives the error probability of each repeated bit, and the errors of the k bits accumulate along the message as in Section 1.6 of Topic 1.

n	R	P_e	Value
1	1	$\frac{1}{10}$	
3	$\frac{1}{3}$	$\frac{7}{250}$	
5	$\frac{1}{5}$	$\frac{107}{12500}$	
11	$\frac{1}{11}$	$\frac{1}{10^{11}} \sum_{i=6}^{11} \binom{11}{i} 9^{11-i}$	$3.0 \cdot 10^{-4}$
21	$\frac{1}{21}$	$\frac{1}{10^{21}} \sum_{i=11}^{21} \binom{21}{i} 9^{21-i}$	$1.4 \cdot 10^{-6}$
23	$\frac{1}{23}$	$\frac{1}{10^{23}} \sum_{i=12}^{23} \binom{23}{i} 9^{23-i}$	$4.7 \cdot 10^{-7}$

Table 2.5: Rate and error probability of the repetition code on the bit-flipping channel with $p = \frac{1}{10}$. The last column gives the size of the long sums.

The capacity of this channel, from the information theory course, is

$$C = 1 - h_2\left(\frac{1}{10}\right) \approx 0.53 \quad \text{bits per channel use,} \quad (2.20)$$

where h_2 is the binary entropy function. Repetition needs $n = 23$ to reach $P_e < 10^{-6}$, at rate $R = \frac{1}{23}$, more than ten times smaller than the capacity. Topic 3 says what that number promises.

On the Gaussian channel there is a second way of looking at the price, in terms of energy. With E_s the mean energy per channel use, a code of rate $R = k/n$ spends on each information bit

$$E_b = \frac{nE_s}{k} = \frac{E_s}{R}. \quad (2.21)$$

For the repetition code with 2-PAM symbols, $E_b = nA^2$. Writing (2.15) in terms of E_b ,

$$P_e = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{nA^2}{2\sigma^2}}\right) = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{2\sigma^2}}\right), \quad (2.22)$$

which is exactly the error probability of uncoded 2-PAM with the same energy per bit, given by equation (1.22) of Topic 1. In other words, at equal energy per bit, the repetition code with soft decisions gains nothing at all. It is equivalent to sending each bit once with n times the energy, and the hard receiver is worse than that. The reduction in error probability is paid entirely with energy and rate.

Not every code is so limited. The **minimum distance** of a code is the smallest Euclidean distance between two of its codewords, and the following problem shows the simplest code whose minimum distance, at equal energy per bit, is larger than that of repetition.

Problem 2.9. Two bits (D_1, D_2) are encoded into three 2-PAM symbols. The encoder sends D_1, D_2 and their sum modulo 2, with $0 \mapsto -A$ and $1 \mapsto +A$. The four messages are equiprobable. Find the rate and the codewords, decode $\mathbf{y} = (1, -\frac{1}{2}, -1)A$ with the ML decoder on the real Gaussian channel, and compare the minimum distance between codewords with that of uncoded 2-PAM and of the repetition code with $n = 3$, at the same energy per bit.

Solution: The rate is $R = 2/3$. The third bit is 0 when $D_1 = D_2$ and 1 otherwise. Table 2.6 lists the four codewords, in units of A , and any two of them differ in exactly two positions. By Proposition 2.3(a), we compute the squared distances from $\mathbf{y}$ to the four codewords, which are in the last column of the table. For the first codeword, for instance, the squared distance is $(2^2 + (\frac{1}{2})^2 + 0^2)A^2 = \frac{17}{4}A^2$.

(D_1, D_2)	Codeword	$\ \mathbf{y} - \mathbf{x}\ ^2$
(0, 0)	$(-, -, -)$	$\frac{17}{4}A^2$
(0, 1)	$(-, +, +)$	$\frac{41}{4}A^2$
(1, 0)	$(+, -, +)$	$\frac{17}{4}A^2$
(1, 1)	$(+, +, -)$	$\frac{9}{4}A^2$

Table 2.6: The four codewords of the parity code and their squared distances to $\mathbf{y} = (1, -\frac{1}{2}, -1)A$.

The decoder decides $(+, +, -)$, that is, $(D_1, D_2) = (1, 1)$. Note that the symbolwise detector would have produced $(+, -, -)$, which is not a codeword. The weak second observation is outvoted by the other two through the parity symbol.

Two codewords that differ in two positions are at squared distance $2 \cdot (2A)^2 = 8A^2$, and this is the squared minimum distance of the code. Figure 2.4 shows the geometry. The energy per bit is $E_b = 3A^2/2$, since three symbols carry two bits. To compare with the other schemes at equal energy per bit, note that scaling every symbol by a factor α multiplies E_b and every squared distance by α^2 , so the ratio $d_{\min}^2/E_b$ does not change and can be computed for each scheme as it stands. Table 2.7 does so.

Scheme	R	E_b	$d_{\min}^2$	$d_{\min}^2/E_b$
Uncoded 2-PAM	1	A^2	$4A^2$	4
Repetition, $n = 3$	$1/3$	$3A^2$	$12A^2$	4
Parity code	$2/3$	$3A^2/2$	$8A^2$	$16/3$

Table 2.7: Rate, energy per bit and squared minimum distance of three schemes with 2-PAM symbols of amplitude A .

At equal energy per information bit, the parity code has $d_{\min}^2 = \frac{16}{3}E_b$, compared with $4E_b$ for the other two schemes. Its squared minimum distance is larger by a factor $\frac{4}{3}$, that is, an asymptotic gain of up to 1.25 dB. At high SNR the error probability is governed by the closest pairs of codewords, as it was governed in Topic 1 by the neighboring points of 4-PAM at distance $2a$. Two codewords at distance d are confused with probability $\frac{1}{2} \operatorname{erfc}(d/(2\sqrt{2}\sigma))$, as Exercise 2.7 shows, and Exercise 2.6 turns this into a bound for the parity code. Better codes make the gain larger, at the price of longer codewords and more work at the decoder.

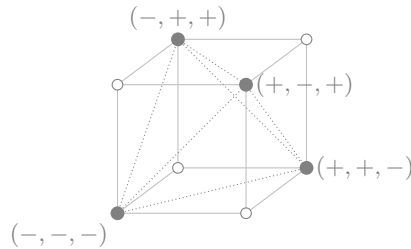


Figure 2.4: The eight sequences of three 2-PAM symbols are the vertices of a cube of side $2A$. The four codewords of the parity code (filled) are pairwise at distance $2\sqrt{2}A$, the diagonal of a face. The empty vertices are the sequences of odd parity.

This leads to the question that Topic 3 answers, namely whether there is a code whose rate stays above a fixed positive value while the error probability goes to zero, and which does better than repetition at equal energy per bit.

Shannon's answer is yes to both. Every channel has a **capacity** C , and for every rate below C there are codes whose error probability is as small as we wish. The capacity is strictly positive for every channel of Table 1.2 of Topic 1, except in useless cases such as the bit-flipping channel with $p = \frac{1}{2}$ or the erasure channel with $p = 1$. Such codes are not repetition codes. Their codewords are spread out in a way that the decoder can exploit, as the parity code of Problem 2.9 begins to show.

What to remember

- A code sends n symbols for every k bits, and its rate is $R = k/n$. The repetition code has $k = 1$ and $R = 1/n$.
- The MAP decoder $\arg \max_{m \in \mathcal{M}} \Pr(m) W(\mathbf{y}|\mathbf{x}(m))$ is optimal, by the proof of Topic 1, and with equiprobable messages it is the ML decoder. On a memoryless channel, the ML decoder maximizes the sum of the scores $\ln W(y_i|x_i)$, and the MAP decoder adds $\ln \Pr(m)$ to that sum.
- ML uses Euclidean distance for Gaussian noise, Hamming distance for bit flips with $p < \frac{1}{2}$, and consistency with unerased bits for erasures. For repetition, these rules become the sign of the sum on the real Gaussian channel, majority voting, and the first unerased bit. With exponential noise, ML selects the feasible codeword with the largest total input energy.
- With the ML decoder, the repetition code has $P_e = \frac{1}{2} \operatorname{erfc}(\sqrt{n \text{SNR}/2})$ on the real Gaussian channel, so n repetitions are worth n times the SNR.
- On the Gaussian channel, hard decisions discard the magnitudes of the observations and lose with respect to soft decisions, up to 3 dB for long repetition codes. On the exponential channel they lose nothing.
- On every channel that carries information, repetition at fixed symbol energy drives P_e to zero, but its rate vanishes too. At equal energy per bit, Gaussian repetition with soft decoding gains nothing over uncoded transmission. Topic 3 shows that better codes exist.

Exercises

The last two exercises are more demanding than the others.

Exercise 2.1. On the bit-erasing channel with $p = \frac{1}{10}$ and equiprobable bits, find the smallest repetition length for $P_e \leq \frac{1}{100}$ and its rate. Compare with the smallest odd repetition length that reaches the same target on the bit-flipping channel in Table 2.5.

Exercise 2.2. Write the error probability of the repetition code with majority decoding on the bit-flipping channel for even n , with ties resolved by a coin, and check that $n = 2$ gives $P_e = p$. What is the ML decoder when $p > \frac{1}{2}$, and when $p = \frac{1}{2}$?

Exercise 2.3. Redo Problem 2.7 with the priors $\Pr(D = 1) = \frac{4}{5}$ and $\Pr(D = 1) = \frac{1}{2}$, and check that in both cases the MAP decoder is the majority vote. For which values of the prior does it stop being so?

Exercise 2.4. On the real Gaussian channel, the bit D has prior $\Pr(D = 1) = q$, with $0 < q < 1$. Show that the MAP decoder of the repetition code decides 1 when $\sum_{i=1}^n y_i$ exceeds a threshold, find the threshold, and compare it with Problem 1.7 of Topic 1.

Exercise 2.5. On the exponential channel, the ML detector of Topic 1 decides $\hat{c}_i = 1$ when $y_i \geq 2A$. Show that, seen from the coded bits to the detected bits, the channel is one in which a 1 is never flipped and a 0 is flipped with probability $e^{-2A/\mu}$. Find the ML decoder of the repetition code on this channel, and show that it makes the same decisions as the decoder of Example 2.4(d).

Exercise 2.6. The parity code of Problem 2.9 is used with the hard receiver: each symbol is detected with the symbolwise detector, whose error probability is $p = \frac{1}{2} \operatorname{erfc}(\sqrt{\text{SNR}/2})$, and the decoder sees three bits. Show that a single wrong bit is noticed, because the three bits have odd parity, but cannot be located, and that two wrong bits go unnoticed. Compute the probabilities that an error is noticed and that it is not. Show that a hard ML decoder, which picks at random one of the three codewords nearest to an odd-parity word, has error probability $2p - p^2$. Finally, use Exercise 2.7 to bound the error probability of the soft decoder by $\frac{3}{2} \operatorname{erfc}(\sqrt{\text{SNR}})$, and compare the three receivers at $\text{SNR} = 2$ without evaluating erfc .

Exercise 2.7. On the real Gaussian channel, a codeword $\mathbf{x}$ is sent and $\mathbf{x}'$ is another codeword at Euclidean distance $d > 0$. Show that the ML decoder finds $\mathbf{x}'$ more likely than $\mathbf{x}$ with probability $\frac{1}{2} \operatorname{erfc}(d/(2\sqrt{2}\sigma))$, by writing the comparison of the two distances in terms of the component of the noise along the line that joins the two codewords. For a unit vector $\mathbf{u}$, the component $\sum_{i=1}^n u_i Z_i$ is a sum of independent Gaussians of variances $u_i^2 \sigma^2$, hence Gaussian of variance σ^2 , as in Problem 2.5. Use equation (1.23) of Topic 1 with threshold $d/2$, and check that the result reduces to (1.22) of Topic 1 for 2-PAM.

Exercise 2.8. For odd n and small p , the sum (2.4) is dominated by its first term, and for large u we have $\operatorname{erfc}(u) \approx e^{-u^2}/(u\sqrt{\pi})$. Use both facts to show that, at high SNR, the error probability of the hard receiver decays like $e^{-(n+1)\text{SNR}/4}$ and that of the soft receiver like $e^{-n\text{SNR}/2}$. Deduce that the hard receiver needs about $2n/(n+1)$ times the SNR of the soft one, that is, 1.8 dB more for $n = 3$ and 3 dB more for large n .

Solutions

Exercise 2.1. By Problem 2.6, $P_e = \frac{1}{2} \cdot 10^{-n}$. At $n = 1$ this is $\frac{1}{20} > \frac{1}{100}$, whereas at $n = 2$ it is $\frac{1}{200} < \frac{1}{100}$. Hence the smallest length is 2, with rate $R = \frac{1}{2}$. On the bit-flipping channel, the odd lengths 1 and 3 fail the target. At $n = 5$, equation (2.4) gives

$$P_e = \frac{10 \cdot 9^2 + 5 \cdot 9 + 1}{10^5} = \frac{107}{12500} < \frac{125}{12500} = \frac{1}{100}. \quad (2.23)$$

Thus the smallest odd length is 5, with rate $R = \frac{1}{5}$. An erasure identifies an unreliable position. A flipped bit looks like a correct one, so the decoder has less information about which positions to trust.

Exercise 2.2. For even n the majority decoder fails when more than $n/2$ bits are flipped, and with probability $\frac{1}{2}$ when exactly $n/2$ are, because the vote is then tied. Hence

$$P_e = \frac{1}{2} \binom{n}{n/2} p^{n/2} (1-p)^{n/2} + \sum_{i=n/2+1}^n \binom{n}{i} p^i (1-p)^{n-i}. \quad (2.24)$$

For $n = 2$ the first term is $\frac{1}{2} \cdot 2p(1-p) = p(1-p)$ and the sum has the single term p^2 , so $P_e = p(1-p) + p^2 = p$, as in Example 2.8. When $p > \frac{1}{2}$, the ratio $p/(1-p)$ in the proof of Proposition 2.3(b) is larger than 1, so the likelihood grows with the Hamming distance and the ML decoder chooses the codeword farthest from $\mathbf{y}$. For the repetition code this is the minority vote, which is the majority vote applied after flipping every received bit. When $p = \frac{1}{2}$, the likelihood is 2^{-n} for every codeword and every $\mathbf{y}$. The output is independent of the input, every decoder is an ML decoder, and with equiprobable bits all of them have $P_e = \frac{1}{2}$.

Exercise 2.3. For $0 < q < 1$, with $\Pr(D = 1) = q$ and d ones in $\mathbf{y}$, the MAP decoder decides 1 when $qp^{3-d}(1-p)^d > (1-q)p^d(1-p)^{3-d}$. Proceeding as in Problem 2.7, with $(1-p)/p = 9$, the condition is

$$\frac{q}{1-q} > 9^{3-2d}. \quad (2.25)$$

The right-hand side takes the values 729, 9, $\frac{1}{9}$ and $\frac{1}{729}$ for $d = 0, 1, 2, 3$. For $q = \frac{4}{5}$ the left-hand side is 4, which is below 9 and above $\frac{1}{9}$, so the decoder decides 0 for $d \leq 1$ and 1 for $d \geq 2$. This is the majority vote. For $q = \frac{1}{2}$ the left-hand side is 1 and the decisions are the same, as they must be, since the MAP decoder is then the ML decoder. The decoder is the majority vote as long as the prior odds lie between $\frac{1}{9}$ and 9, that is, for $\frac{1}{10} < q < \frac{9}{10}$. For $q > \frac{9}{10}$ it decides 1 with a single received 1, as in Problem 2.7, and for $q > \frac{729}{730}$ it decides 1 whatever it receives. For $q < \frac{1}{10}$ the same happens in favor of 0, with a constant decision for $q < \frac{1}{730}$. At the four boundary values the two options tie, and either choice is a MAP decision.

Exercise 2.4. The MAP decoder decides 1, that is, the codeword $(+A, \dots, +A)$, when $q \prod_{i=1}^n W(y_i | +A) > (1-q) \prod_{i=1}^n W(y_i | -A)$. Taking logarithms and canceling the common factors, as in Problem 1.7 of Topic 1, the condition is

$$-\sum_{i=1}^n \frac{(y_i - A)^2}{2\sigma^2} + \ln q > -\sum_{i=1}^n \frac{(y_i + A)^2}{2\sigma^2} + \ln(1-q). \quad (2.26)$$

Expanding the squares, the terms in y_i^2 and A^2 cancel, and what remains is linear in the observations,

$$\sum_{i=1}^n y_i > \frac{\sigma^2}{2A} \ln \frac{1-q}{q}. \quad (2.27)$$

The threshold is the same as in Problem 1.7 of Topic 1, but it is now applied to the sum of the n observations. In terms of their average, the threshold is divided by n . The prior therefore weighs less and less as n grows, because the n observations together carry more evidence than a single one.

Exercise 2.5. When the coded bit is 1 the symbol $2A$ is sent, and the output $y_i = 2A + Z_i$ is at least $2A$ because the noise is nonnegative. The detector decides $\hat{c}_i = 1$ and a 1 is never flipped. When the coded bit is 0 the output is the noise alone, and the detector decides 1 when $Z_i \geq 2A$, which by Example 1.14 of Topic 1 has probability $\varepsilon = e^{-2A/\mu}$. For the repetition code, let d be the number of ones among the n detected bits. If $D = 1$ is sent, all the detected bits are 1 with probability one, so the likelihood of the detected sequence is 1

when $d = n$ and 0 otherwise. If $D = 0$ is sent, the likelihood is $\varepsilon^d(1 - \varepsilon)^{n-d}$. Hence, when some detected bit is 0 only $D = 0$ is possible, and when all of them are 1 the likelihood of $D = 1$ is 1, larger than ε^n . The ML decoder decides 1 when all the detected bits are 1, that is, when every $y_i \geq 2A$, and 0 otherwise. This is the decoder of Example 2.4(d), so on this channel hard decisions lose nothing.

Exercise 2.6. In bits, the four codewords are 000, 011, 101 and 110, the sequences of three bits with an even number of ones. A single wrong bit turns a codeword into a sequence of odd parity, which is not a codeword, so the error is noticed. It cannot be located, because every sequence of odd parity is at Hamming distance 1 from three different codewords. Two wrong bits turn a codeword into another codeword, so the error goes unnoticed, and three wrong bits give odd parity again. With independent errors of probability p , an error is noticed with probability

$$3p(1-p)^2 + p^3, \quad (2.28)$$

and goes unnoticed with probability

$$3p^2(1-p), \quad (2.29)$$

so the hard receiver fails in one way or the other with probability $1 - (1-p)^3$. The hard ML decoder guesses right at an odd-parity word with probability $\frac{1}{3}$ when one bit is wrong, since the transmitted codeword is one of the three nearest, and never when three bits are wrong. Its probability of a correct decision is therefore $(1-p)^3 + p(1-p)^2 = (1-p)^2$, and its error probability is

$$P_e = 1 - (1-p)^2 = 2p - p^2. \quad (2.30)$$

For the soft decoder, each codeword has three neighbors at the minimum distance $d = 2\sqrt{2}A$. By Exercise 2.7, the decoder prefers a given neighbor with probability $\frac{1}{2} \operatorname{erfc}(d/(2\sqrt{2}\sigma)) = \frac{1}{2} \operatorname{erfc}(A/\sigma) = \frac{1}{2} \operatorname{erfc}(\sqrt{\text{SNR}})$, and the probability that at least one of the three events occurs is at most the sum of their probabilities,

$$P_e \leq \frac{3}{2} \operatorname{erfc}(\sqrt{\text{SNR}}). \quad (2.31)$$

At $\text{SNR} = 2$, with $a = \frac{1}{2} \operatorname{erfc}(1)$ the error probability of one detected bit, the hard receiver fails with probability $1 - (1-a)^3$, the hard ML decoder errs with probability $2a - a^2$, and the soft decoder with probability at most $\frac{3}{2} \operatorname{erfc}(\sqrt{2})$. The first two differ by $a(1-a)^2$, the cases in which the decoder guesses right where the receiver only notices the error. The soft decoder is the ML decoder of the whole observation, so by Theorem 2.2 its error probability is at most $2a - a^2$, whatever the value of the bound.

Exercise 2.7. With $\mathbf{y} = \mathbf{x} + \mathbf{z}$, the decoder finds $\mathbf{x}'$ more likely than $\mathbf{x}$ when $\|\mathbf{y} - \mathbf{x}'\|^2 < \|\mathbf{y} - \mathbf{x}\|^2$, that is, when $\|\mathbf{z} - (\mathbf{x}' - \mathbf{x})\|^2 < \|\mathbf{z}\|^2$. Expanding the square on the left,

$$\|\mathbf{z}\|^2 - 2 \sum_{i=1}^n z_i(x'_i - x_i) + d^2 < \|\mathbf{z}\|^2, \quad (2.32)$$

where $d = \|\mathbf{x}' - \mathbf{x}\|$. With the unit vector $\mathbf{u} = (\mathbf{x}' - \mathbf{x})/d$, which points from $\mathbf{x}$ to $\mathbf{x}'$, the condition becomes

$$\sum_{i=1}^n u_i z_i > \frac{d}{2}. \quad (2.33)$$

Only the component of the noise along the line that joins the two codewords matters, and this competitor beats the transmitted word when the component exceeds half the distance. With other codewords present, the final decision need not be this particular competitor. The component is Gaussian with zero mean and variance $\sum_{i=1}^n u_i^2 \sigma^2 = \sigma^2$, so by equation (1.23) of Topic 1 with threshold $d/2$ the probability is $\frac{1}{2} \operatorname{erfc}(d/(2\sqrt{2}\sigma))$. For 2-PAM the two points are at distance $d = 2A$, and the result is $\frac{1}{2} \operatorname{erfc}(A/(\sqrt{2}\sigma)) = \frac{1}{2} \operatorname{erfc}(\sqrt{\text{SNR}/2})$, which is (1.22) of Topic 1.

Exercise 2.8. For the hard receiver, the first term of (2.4) is $\binom{n}{(n+1)/2} p^{(n+1)/2} (1-p)^{(n-1)/2}$, and at high SNR the factor that decides how fast it decays is $p^{(n+1)/2}$. By (2.19) and the approximation of erfc , the probability p decays like $e^{-\text{SNR}/2}$, so the error probability of the hard receiver decays like

$$(e^{-\text{SNR}/2})^{(n+1)/2} = e^{-(n+1)\text{SNR}/4}. \quad (2.34)$$

For the soft receiver, (2.15) and the same approximation give a decay like $e^{-n \text{SNR}/2}$. The two receivers reach the same exponent when $(n+1)\text{SNR}_{\text{hard}}/4 = n \text{SNR}_{\text{soft}}/2$, that is, when

$$\frac{\text{SNR}_{\text{hard}}}{\text{SNR}_{\text{soft}}} = \frac{2n}{n+1}. \quad (2.35)$$

For $n = 3$ the ratio is $\frac{3}{2}$, or 1.8 dB, and for large n it tends to 2, or 3 dB. The hard receiver needs $(n+1)/2$ wrong bits to fail, each of them paid with the exponent $\text{SNR}/2$, whereas the soft receiver has to overcome the energy of all n symbols at once.