1
Error-correcting Codes

Ruud Pellikaan and Xin-Wen Wu

The idea of *redundant* information is a well-known phenomenon in reading a newspaper. Misspellings usually go unnoticed for a casual reader, while the meaning is still grasped. In Semitic languages such as Hebrew, and even older in the hieroglyphics in the tombs of the pharaohs of Egypt, only the consonants are written while the vowels are left out so that we do not know for sure how to pronounce these words nowadays. The letter "e" is the most frequent occurring symbol in the English language, and leaving out all these letters would still give in almost all cases an understandable text to the expense of greater attention of the reader.

The science of deleting redundant information in a clever way such that it can be stored in less memory or space and still can be expanded to the original message is called *data compression* or *source coding*. It is not the topic of this book. So we can compress data, but an error made in a compressed text would give a different message that is most of the time utterly meaningless.

The idea in *error-correcting codes* is the converse. One adds redundant information in such a way that it is possible to detect or even correct errors after transmission. In radio contacts between pilots and radar controls the letters in the alphabet are spoken phonetically as "Alpha, Bravo, Charlie, ..." but "Adams, Boston, Chicago, ..." is more commonly used for spelling in a telephone conversation. The addition of a parity check symbol enables one to detect an error, such as on the former punch cards that were fed into a computer, in the ISBN code for books, the European Article Numbering (EAN) and the Universal Product Code (UPC) for articles. Error-correcting codes are common in numerous modern applications where data are required to be stored, processed and transmitted in a reliable manner, such as in audiovisual media, quick response (QR) codes, fault-tolerant computer systems and deep space telecommunication, to name but a few.

In this chapter, we present an introduction to error-correcting codes focusing on the most commonly used codes, namely, block codes and linear codes, including fundamental concepts and procedures for code construction, encoding and decoding.

1.1 Block Codes

Legend goes that Hamming was so frustrated the computer halted every time it detected an error after he handed in a stack of punch cards, he thought about a way the computer would be able not only to detect the error but also to correct it automatically. He came up with his nowadays famous code named after him. Whereas the theory of Hamming is about the actual construction, the encoding and decoding of codes and uses tools from *combinatorics* and *algebra*, the approach of Shannon leads to *information theory* and his theorems tell us what is and what is not possible in a *probabilistic* sense.

According to Shannon we have a message $\mathbf{m}$ in a certain alphabet and of a certain length, we encode $\mathbf{m}$ to $\mathbf{c}$ by expanding the length of the message and adding redundant information. One can define the *information rate* R that measures the slowing down of the transmission of the data. The encoded message $\mathbf{c}$ is sent over a noisy channel such that the symbols are changed, according to certain probabilities that are characteristic of the channel. The received word $\mathbf{r}$ is decoded to $\mathbf{m}'$. See Figure 1.1. Now given the characteristics of the channel one can define the *capacity* C of the channel and it has the property that for every $R < C$ it is possible to find an encoding and decoding scheme such that the *error probability* that $\mathbf{m}' \neq \mathbf{m}$ is arbitrarily small. For $R > C$ such

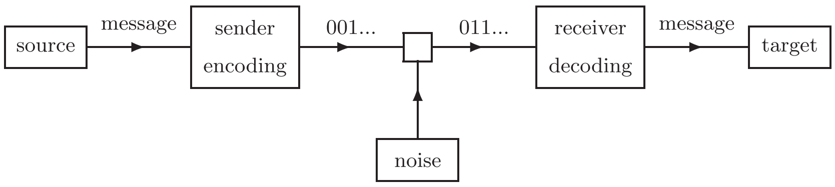

Figure 1.1 Block diagram of a communication system

a scheme is not possible. The capacity is explicitly known as a function of the characteristic probability for quite a number of channels.

The notion of a channel must be taken in a broad sense. Not only the transmission of data via satellite or telephone but also the storage of information on a hard disk of a computer or a compact disc for music and film can be modelled by a channel.

The theorem of Shannon tells us the existence of certain encoding and decoding schemes and one can even say that they exist in abundance and that almost all schemes satisfy the required conditions, but it does not tell us how to construct a specific scheme efficiently. The information theoretic part of error-correcting codes is considered in this book only so far as to motivate the construction of coding and decoding algorithms.

The situation for the best codes in terms of the maximal number of errors that one can correct for a given information rate and code length is not so clear. Several existence and nonexistence theorems are known, but the exact bound is in fact still an open problem.

1.1.1 Repetition, Product and Hamming Codes

Adding a parity check such that the number of ones is even, is a well-known way to detect one error. But this does not correct the error.

Example 1.1.1 Replacing every symbol by a threefold repetition gives the possibility of correcting one error in every 3-tuple of symbols in a received word by a majority vote. The price one has to pay is that the transmission is three times slower. We say that the *information rate* is 1/3. We see here the two conflicting demands of error-correction: to correct as many errors as possible and to transmit as fast as possible. Notice furthermore that in case two errors are introduced by transmission the majority decoding rule will introduce a *decoding error* in this example.

Example 1.1.2 An improvement is the following product construction. Suppose we want to transmit a binary message (m_1, m_2, m_3, m_4) of length 4 by adding 5 redundant bits $(r_1, r_2, r_3, r_4, r_5)$. Put these 9 bits in a 3×3 array as shown below. The redundant bits are defined by the following conditions. The sum of the number of ones in every row and in every column should be even.

m_1	m_2	r_1
m_3	m_4	r_2
r_3	r_4	r_5

It is clear that r_1, r_2, r_3 and r_4 are well defined by these rules. The condition on the last row and on the last column are equivalent, given the rules for the first two rows and columns. Hence r_5 is also well defined. Suppose that the message is $\mathbf{m} = (1,1,0,1)$. Then the redundant part is $\mathbf{r} = (0,1,1,0,1)$ and $\mathbf{c} = (1,1,0,1,0,1,1,0,1)$ is transmitted.

If by the transmission 3 bits are erased, that means that the receiver knows the positions of the *erasures* but not its values, then the receiver can fill in the blanks. Suppose that $\mathbf{y} = (1,-,0,-,0,-,1,0,1)$ is the received word.

1		0	$\leftarrow$
0			
1	0	1	
		$\uparrow$	

The number of ones in every row and column should be even, so the receiver fills a 1 at the blank in the first row and the last column, and consequently a 1 in the middle.

If in the transmission of the word of 9 bits, one symbol is flipped from 0 to 1 or vice versa, then the receiver will notice this, and is able to correct it. Since if the error occurred in row i and column j, then the receiver will detect an odd parity in this row and this column and an even parity in the remaining rows and columns. Suppose that $\mathbf{y} = (1,1,0,1,0,0,1,0,1)$ is the received word.

1	1	0	
0	1	0	$\leftarrow$
1	0	1	
		$\uparrow$	

Then the receiver detects an error in row 2 and column 3 and will change the corresponding symbol.

So this product code can also correct one error as the repetition code but its information rate is improved from 1/3 to 4/9.

This decoding scheme is *incomplete* in the sense that in some cases it is not decided what to do and the scheme will fail to determine a candidate for the transmitted word. That is called a *decoding failure*. Sometimes two errors can be corrected. If the first error is in row i and column j, and the second in row i' and column j' with $i' > i$ and $j' \neq j$, then the receiver will detect odd parities in rows i and i' and in columns j and j'. There are two error patterns of two errors with this behavior. That is errors at the positions (i, j) and (i', j') or at the two pairs (i, j') and (i', j). If the receiver decides to change the first two pairs if $j' > j$ and the second two pairs if $j' < j$, then it will recover the transmitted word half of the time this pattern of two errors takes place. If for instance the word $\mathbf{c} = (1, 1, 0, 1, 0, 1, 1, 0, 1)$ is transmitted and $\mathbf{y} = (1, 0, 0, 1, 0, 0, 1, 0, 1)$ is received, then the above decoding scheme will change it correctly in $\mathbf{c}$. But if $\mathbf{y}' = (1, 1, 0, 0, 1, 1, 1, 0, 1)$ is received, then the scheme will change it in the codeword $\mathbf{c}' = (1, 0, 0, 0, 1, 0, 1, 0, 1)$ and we have a decoding error.

1	0	0	$\leftarrow$
0	1	0	$\leftarrow$
1	0	1	
	$\uparrow$	$\uparrow$	

1	1	1	$\leftarrow$
0	0	1	$\leftarrow$
1	0	1	
	$\uparrow$	$\uparrow$	

If two errors take place in the same row, then the receiver will see an even parity in all rows and odd parities in the columns j and j'. We can expand the decoding rule to change the bits at the positions $(1, j)$ and $(1, j')$. Likewise we will change the bits in positions $(i, 1)$ and $(i', 1)$ if the columns give even parity and the rows i and i' have an odd parity. This decoding scheme will correct all patterns with one error correctly, and sometimes the patterns with two errors. But it is still incomplete, since the received word $(1, 1, 0, 1, 1, 0, 0, 1, 0)$ has an odd parity in every row and in every column and the scheme fails to decode.

One could extend the decoding rule to get a complete decoding in such a way that every received word is decoded to a nearest codeword. This nearest codeword is not always unique.

In case the transmission is by means of certain electromagnetic pulses or waves one has to consider *modulation* and *demodulation*. The message consists of letters of a finite alphabet, say consisting of zeros and ones, and these are modulated, transmitted as waves, received and demodulated in zeros and ones. In the demodulation part one has to make a *hard decision* between a zero or a one. But usually there is a probability that the signal represents a zero. The hard decision together with this probability is called a *soft decision*. One can make use of this information in the decoding algorithm. One considers the *list* of all nearest codewords, and one chooses the codeword in this list that has the highest probability.

Example 1.1.3 An improvement of the repetition code of rate $1/3$ and the product code of rate $4/9$ is given by Hamming. Suppose we have a message (m_1, m_2, m_3, m_4) of 4 bits. Put them in the middle of the following *Venn diagram* of three intersecting circles as given in Figure 1.2. Complete the three empty areas of the circles according to the rule that the number of ones in every circle is even. In this way we get 3 redundant bits (r_1, r_2, r_3) that we add to the message and which we transmit over the channel.

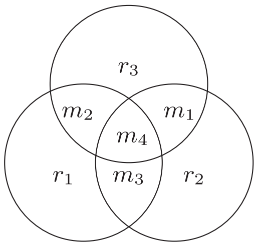

Figure 1.2 Venn diagram of the Hamming code

In every block of 7 bits the receiver can correct one error. Since the parity in every circle should be even. So if the parity is even we declare the circle correct, if the parity is odd we declare the circle incorrect. The error is in the incorrect circles and in the complement of the correct circles. We see that every pattern of at most one error can be corrected in this way. For instance, if $\mathbf{m} = (1, 1, 0, 1)$ is the message, then $\mathbf{r} = (0, 0, 1)$ is the redundant information added and $\mathbf{c} = (1, 1, 0, 1, 0, 0, 1)$ the codeword sent. If after transmission one symbol is flipped and $\mathbf{y} = (1, 0, 0, 1, 0, 0, 1)$ is the received word as given in Figure 1.3, then we conclude that the error is in the left and upper circle, but not in the right one.

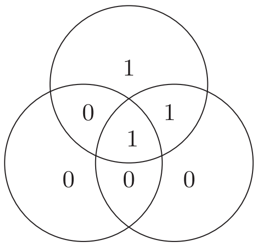

Figure 1.3 Venn diagram of a received word for the Hamming code

And we conclude that the error is at m_2. But in case of two errors and for instance the word $\mathbf{y}' = (1, 0, 0, 1, 1, 0, 1)$ is received, then the receiver would assume that the error occurred in the upper circle and not in the two lower circles, and would therefore conclude that the transmitted codeword was $(1,0,0,1,1,0,0)$. Hence the decoding scheme creates an extra error.

The redundant information $\mathbf{r}$ can be obtained from the message $\mathbf{m}$ by means of three linear equations or parity checks modulo two

$$\begin{cases} r_1 = m_2 + m_3 + m_4 \\ r_2 = m_1 + m_3 + m_4 \\ r_3 = m_1 + m_2 + m_4. \end{cases}$$

Let $\mathbf{c} = (\mathbf{m}, \mathbf{r})$ be the codeword. Then $\mathbf{c}$ is a codeword if and only if $H\mathbf{c}^T = 0$, where

$$H = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

The information rate is improved from $1/3$ for the repetition code and $4/9$ for the product code to $4/7$ for the Hamming code.

1.1.2 Codes and Hamming Distance

In general the alphabets of the message word and the encoded word might be distinct. Furthermore the length of both the message word and the encoded word might vary such as in a *convolutional code*. We restrict ourselves to $[n,k]$ block codes that is the message words have a fixed length of k symbols and the encoded words have a fixed length of n symbols both from the same alphabet Q. For the purpose of error control, before transmission, we add redundant symbols to the message in a clever way.

Definition 1.1.4 Let Q be a set of q symbols called the *alphabet*. Let Q^n be the set of all n-tuples $\mathbf{x} = (x_1, \ldots, x_n)$, with entries $x_i \in Q$. A *block code* C of *length* n over Q is a nonempty subset of Q^n. The elements of C are called *codewords*. If C contains M codewords, then M is called the *size* of the code. We call a code with length n and size M an (n,M) code. If $M = q^k$, then C is called an $[n,k]$ code. For an (n,M) code defined over Q, the value $n - \log_q(M)$ is called the *redundancy*. The *information rate* is defined as $R = \log_q(M)/n$.

Example 1.1.5 The repetition code has length 3 and 2 codewords, so its information rate is $1/3$. The product code has length 9 and 2^4

codewords, hence its rate is $4/9$. The Hamming code has length 7 and 2^4 codewords, therefore its rate is $4/7$.

Example 1.1.6 Let C be the binary block code of length n consisting of all words with exactly two ones. This is an $(n, n(n-1)/2)$ code. In this example the number of codewords is not a power of the size of the alphabet.

Definition 1.1.7 Let C be an $[n, k]$ block code over Q. An *encoder* of C is a one-to-one map

$$\mathcal{E} : Q^k \longrightarrow Q^n$$

such that $C = \mathcal{E}(Q^k)$. Let $\mathbf{c} \in C$ be a codeword. Then there exists a unique $\mathbf{m} \in Q^k$ with $\mathbf{c} = \mathcal{E}(\mathbf{m})$. This $\mathbf{m}$ is called the *message* or *source word* of $\mathbf{c}$.

In order to measure the difference between two distinct words and to evaluate the error-correcting capability of the code, we need to introduce an appropriate metric to Q^n. A natural metric used in Coding Theory is the *Hamming distance.*

Definition 1.1.8 For $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_n) \in Q^n$, the Hamming distance $d(\mathbf{x}, \mathbf{y})$ is defined as the number of places where they differ:

$$d(\mathbf{x}, \mathbf{y}) = |\{i \mid x_i \neq y_i\}|.$$

Proposition 1.1.9 *The Hamming distance is a well-defined metric on Q^n, that means that the following properties hold for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in Q^n$:*
(1) $d(\mathbf{x}, \mathbf{y}) \geq 0$, *and equality holds if and only if* $\mathbf{x} = \mathbf{y}$*;*
(2) $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ (symmetry)*;*
(3) $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ (triangle inequality).
The triangle inequality is shown in Figure 1.4.

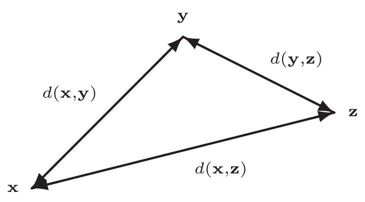

Figure 1.4 Triangle inequality

Proof. Properties (1) and (2) are trivial from the definition. We leave (3) to the reader as an exercise. ◇

Definition 1.1.10 The *minimum distance* of a code $C \subseteq Q^n$ is defined as

$$d = d(C) = \min\{\ d(\mathbf{x}, \mathbf{y}) \mid \mathbf{x}, \mathbf{y} \in C,\ \mathbf{x} \neq \mathbf{y}\ \}$$

if C consists of more than one element, and is by definition $n + 1$ if C consists of one word. We denote by (n, M, d) the parameters of a code C with length n, size M and minimum distance d.

The main problem of error-correcting codes from "Hamming's point of view" is to construct for given length and number of codewords a code with the largest possible minimum distance, and to find efficient encoding and decoding algorithms for such a code.

Example 1.1.11 The triple repetition code consists of two codewords: $(0, 0, 0)$ and $(1, 1, 1)$, so its minimum distance is 3. The product and Hamming code both correct one error. So the minimum distance is at least 3, by the triangle inequality. The product code has minimum distance 4 and the Hamming code has minimum distance 3. Notice that all three codes have the property that $\mathbf{x} + \mathbf{y}$ is again a codeword if $\mathbf{x}$ and $\mathbf{y}$ are codewords.

Definition 1.1.12 Let $\mathbf{x} \in Q^n$. The *ball of radius r around* $\mathbf{x}$, denoted by $B_r(\mathbf{x})$, is defined by $B_r(\mathbf{x}) = \{\ \mathbf{y} \in Q^n \mid d(\mathbf{x}, \mathbf{y}) \leq r\ \}$. The *sphere of radius r around* $\mathbf{x}$ is denoted by $S_r(\mathbf{x})$ and defined by $S_r(\mathbf{x}) = \{\ \mathbf{y} \in Q^n \mid d(\mathbf{x}, \mathbf{y}) = r\ \}$.

Figure 1.5 shows the ball in the Euclidean plane. This is misleading in some respects, but gives an indication of what we should have in mind. Figure 1.6 shows Q^2, where the alphabet Q consists of 5 elements. The ball $B_0(\mathbf{x})$ consists of the points in the circle, $B_1(\mathbf{x})$ is depicted by the points inside the cross and $B_2(\mathbf{x})$ consists of all 25 dots.

Proposition 1.1.13 *Let Q be an alphabet of q elements and $\mathbf{x} \in Q^n$. Then*

$$|S_i(\mathbf{x})| = \binom{n}{i}(q-1)^i \quad \textit{and} \quad |B_r(\mathbf{x})| = \sum_{i=0}^{r} \binom{n}{i}(q-1)^i.$$

Proof. Let $\mathbf{y} \in S_i(\mathbf{x})$. Let I be the subset of $\{1, \ldots, n\}$ consisting of all positions j such that $y_j \neq x_j$. Then the number of elements of I is

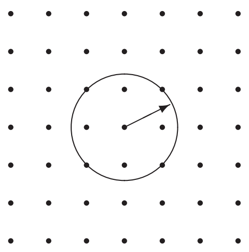

Figure 1.5 Ball of radius $\sqrt{2}$ in the Euclidean plane

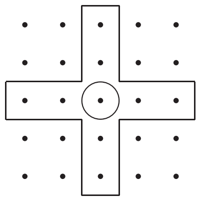

Figure 1.6 Balls of radius 0 and 1 in the Hamming metric

equal to i. And $(q-1)^i$ is the number of words $\mathbf{y} \in S_i(\mathbf{x})$ that have the same fixed I. The number of possibilities to choose the subset I with a fixed number of elements i is equal to $\binom{n}{i}$. This shows the formula for the number of elements of $S_i(\mathbf{x})$.
Furthermore $B_r(\mathbf{x})$ is the disjoint union of the subsets $S_i(\mathbf{x})$ for $i = 0, \ldots, r$. This proves the statement about the number of elements of $B_r(\mathbf{x})$. ◇

Exercises

1.1.1 Consider the code of length 8 that is obtained by deleting the last entry r_5 from the product code of Example 1.1.2. Show that this code corrects one error.

1.1.2 Let C be the code of all $n \times n$ binary arrays that have an even number of ones in every row and column. What is the number of codewords of C? What is the minimum distance of C? Show that every pattern of three erasures can be corrected.

1.1.3 Let $Q = \{1, 2, \ldots, n\}$. Let C be the code in Q^n of all $(c_1, c_2, \ldots, c_n)$ such that $\{c_1, c_2, \ldots, c_n\} = Q$. What is the number of codewords of C? What is the minimum distance of C? Show that for every received word with one error there are exactly two nearest codewords.