Fundamentals of Error-Correcting Codes

W. Cary Huffman

Loyola University of Chicago

and

Vera Pless

University of Illinois at Chicago

CAMBRIDGE UNIVERSITY PRESS

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE
The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS
The Edinburgh Building, Cambridge CB2 2RU, UK
40 West 20th Street, New York, NY 10011-4211, USA
477 Williamstown Road, Port Melbourne, VIC 3207, Australia
Ruiz de Alarcón 13, 28014 Madrid, Spain
Dock House, The Waterfront, Cape Town 8001, South Africa

http://www.cambridge.org

First published 2003

Printed in the United Kingdom at the University Press, Cambridge

Typefaces Times 10/13 pt and Helvetica Neue   *System* LaTeX 2_ε [TB]

A catalog record for this book is available from the British Library

Library of Congress Cataloging in Publication Data

Huffman, W. C. (William Cary)
Fundamentals of error-correcting codes / W. Cary Huffman, Vera Pless.
    p.   cm.
Includes bibliographical references and index.
ISBN 0-521-78280-5
1. Error-correcting codes (Information theory) I. Pless, Vera. II. Title.
QA268 .H84 2003
005.7′2 – dc21 2002067236

ISBN 0 521 78280 5 hardback

Contents

1 Basic concepts of linear codes

In 1948 Claude Shannon published a landmark paper "A mathematical theory of communication" [306] that signified the beginning of both information theory and coding theory. Given a communication channel which may corrupt information sent over it, Shannon identified a number called the capacity of the channel and proved that arbitrarily reliable communication is possible at any rate below the channel capacity. For example, when transmitting images of planets from deep space, it is impractical to retransmit the images. Hence if portions of the data giving the images are altered, due to noise arising in the transmission, the data may prove useless. Shannon's results guarantee that the data can be encoded before transmission so that the altered data can be decoded to the specified degree of accuracy. Examples of other communication channels include magnetic storage devices, compact discs, and any kind of electronic communication device such as cellular telephones.

The common feature of communication channels is that information is emanating from a source and is sent over the channel to a receiver at the other end. For instance in deep space communication, the message source is the satellite, the channel is outer space together with the hardware that sends and receives the data, and the receiver is the ground station on Earth. (Of course, messages travel from Earth to the satellite as well.) For the compact disc, the message is the voice, music, or data to be placed on the disc, the channel is the disc itself, and the receiver is the listener. The channel is "noisy" in the sense that what is received is not always the same as what was sent. Thus if binary data is being transmitted over the channel, when a 0 is sent, it is hopefully received as a 0 but sometimes will be received as a 1 (or as unrecognizable). Noise in deep space communications can be caused, for example, by thermal disturbance. Noise in a compact disc can be caused by fingerprints or scratches on the disc. The fundamental problem in coding theory is to determine what message was sent on the basis of what is received.

A communication channel is illustrated in Figure 1.1. At the source, a message, denoted $\mathbf{x}$ in the figure, is to be sent. If no modification is made to the message and it is transmitted directly over the channel, any noise would distort the message so that it is not recoverable. The basic idea is to embellish the message by adding some redundancy to it so that hopefully the received message is the original message that was sent. The redundancy is added by the encoder and the embellished message, called a codeword $\mathbf{c}$ in the figure, is sent over the channel where noise in the form of an error vector $\mathbf{e}$ distorts the codeword producing a received vector $\mathbf{y}$.[1] The received vector is then sent to be decoded where the errors are

[1] Generally our codeword symbols will come from a field $\mathbb{F}_q$, with q elements, and our messages and codewords will be vectors in vector spaces $\mathbb{F}_q^k$ and $\mathbb{F}_q^n$, respectively; if $\mathbf{c}$ entered the channel and $\mathbf{y}$ exited the channel, the difference $\mathbf{y} - \mathbf{c}$ is what we have termed the error $\mathbf{e}$ in Figure 1.1.

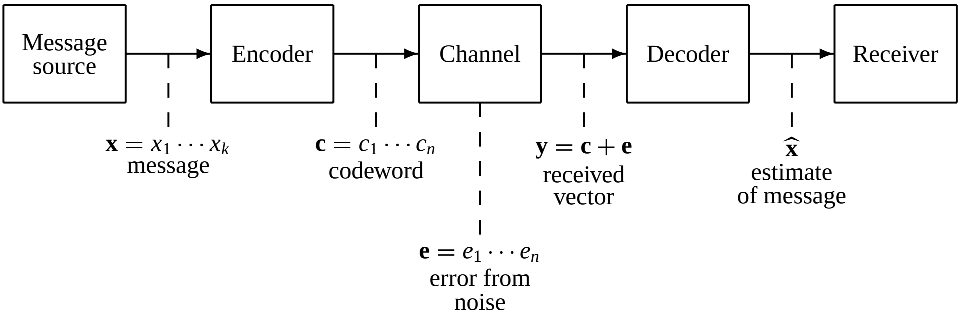

Figure 1.1 Communication channel.

removed, the redundancy is then stripped off, and an estimate $\widehat{\mathbf{x}}$ of the original message is produced. Hopefully $\widehat{\mathbf{x}} = \mathbf{x}$. (There is a one-to-one correspondence between codewords and messages. Thus we will often take the point of view that the job of the decoder is to obtain an estimate $\widehat{\mathbf{y}}$ of $\mathbf{y}$ and hope that $\widehat{\mathbf{y}} = \mathbf{c}$.) Shannon's Theorem guarantees that our hopes will be fulfilled a certain percentage of the time. With the right encoding based on the characteristics of the channel, this percentage can be made as high as we desire, although not 100%.

The proof of Shannon's Theorem is probabilistic and nonconstructive. In other words, no specific codes were produced in the proof that give the desired accuracy for a given channel. Shannon's Theorem only guarantees their existence. The goal of research in coding theory is to produce codes that fulfill the conditions of Shannon's Theorem. In the pages that follow, we will present many codes that have been developed since the publication of Shannon's work. We will describe the properties of these codes and on occasion connect these codes to other branches of mathematics. Once the code is chosen for application, encoding is usually rather straightforward. On the other hand, decoding efficiently can be a much more difficult task; at various points in this book we will examine techniques for decoding the codes we construct.

1.1 Three fields

Among all types of codes, linear codes are studied the most. Because of their algebraic structure, they are easier to describe, encode, and decode than nonlinear codes. The code alphabet for linear codes is a finite field, although sometimes other algebraic structures (such as the integers modulo 4) can be used to define codes that are also called "linear."

In this chapter we will study linear codes whose alphabet is a field $\mathbb{F}_q$, also denoted GF(q), with q elements. In Chapter 3, we will give the structure and properties of finite fields. Although we will present our general results over arbitrary fields, we will often specialize to fields with two, three, or four elements.

A field is an algebraic structure consisting of a set together with two operations, usually called addition (denoted by $+$) and multiplication (denoted by $\cdot$ but often omitted), which satisfy certain axioms. Three of the fields that are very common in the study of linear codes are the *binary* field with two elements, the *ternary* field with three elements, and the *quaternary* field with four elements. One can work with these fields by knowing their addition and multiplication tables, which we present in the next three examples.

Example 1.1.1 The binary field $\mathbb{F}_2$ with two elements $\{0, 1\}$ has the following addition and multiplication tables:

$+$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

This is also the ring of integers modulo 2. ■

Example 1.1.2 The ternary field $\mathbb{F}_3$ with three elements $\{0, 1, 2\}$ has addition and multiplication tables given by addition and multiplication modulo 3:

$+$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

$\cdot$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

■

Example 1.1.3 The quaternary field $\mathbb{F}_4$ with four elements $\{0, 1, \omega, \overline{\omega}\}$ is more complicated. It has the following addition and multiplication tables; $\mathbb{F}_4$ is not the ring of integers modulo 4:

$+$	0	1	ω	$\overline{\omega}$
0	0	1	ω	$\overline{\omega}$
1	1	0	$\overline{\omega}$	ω
ω	ω	$\overline{\omega}$	0	1
$\overline{\omega}$	$\overline{\omega}$	ω	1	0

$\cdot$	0	1	ω	$\overline{\omega}$
0	0	0	0	0
1	0	1	ω	$\overline{\omega}$
ω	0	ω	$\overline{\omega}$	1
$\overline{\omega}$	0	$\overline{\omega}$	1	ω

Some fundamental equations are observed in these tables. For instance, one notices that $x + x = 0$ for all $x \in \mathbb{F}_4$. Also $\overline{\omega} = \omega^2 = 1 + \omega$ and $\omega^3 = \overline{\omega}^3 = 1$. ■

1.2 Linear codes, generator and parity check matrices

Let $\mathbb{F}_q^n$ denote the vector space of all n-tuples over the finite field $\mathbb{F}_q$. An (n, M) *code* $\mathcal{C}$ over $\mathbb{F}_q$ is a subset of $\mathbb{F}_q^n$ of size M. We usually write the vectors $(a_1, a_2, \ldots, a_n)$ in $\mathbb{F}_q^n$ in the form $a_1a_2 \cdots a_n$ and call the vectors in $\mathcal{C}$ *codewords*. Codewords are sometimes specified in other ways. The classic example is the polynomial representation used for codewords in cyclic codes; this will be described in Chapter 4. The field $\mathbb{F}_2$ of Example 1.1.1 has had a very special place in the history of coding theory, and codes over $\mathbb{F}_2$ are called *binary codes*. Similarly codes over $\mathbb{F}_3$ are termed *ternary codes*, while codes over $\mathbb{F}_4$ are called *quaternary codes*. The term "quaternary" has also been used to refer to codes over the ring $\mathbb{Z}_4$ of integers modulo 4; see Chapter 12.

Without imposing further structure on a code its usefulness is somewhat limited. The most useful additional structure to impose is that of linearity. To that end, if $\mathcal{C}$ is a k-dimensional subspace of $\mathbb{F}_q^n$, then $\mathcal{C}$ will be called an $[n, k]$ *linear code* over $\mathbb{F}_q$. The linear code $\mathcal{C}$ has q^k codewords. The two most common ways to present a linear code are with either a generator matrix or a parity check matrix. A *generator matrix* for an $[n, k]$ code $\mathcal{C}$ is any $k \times n$ matrix G whose rows form a basis for $\mathcal{C}$. In general there are many generator matrices for a code. For any set of k independent columns of a generator matrix G, the corresponding set of coordinates forms an *information set* for $\mathcal{C}$. The remaining $r = n - k$ coordinates are termed a *redundancy set* and r is called the *redundancy* of $\mathcal{C}$. If the first k coordinates form an information set, the code has a unique generator matrix of the form $[I_k \mid A]$ where I_k is the $k \times k$ identity matrix. Such a generator matrix is in *standard form*. Because a linear code is a subspace of a vector space, it is the kernel of some linear transformation. In particular, there is an $(n - k) \times n$ matrix H, called a *parity check matrix* for the $[n, k]$ code $\mathcal{C}$, defined by

$$\mathcal{C} = \left\{\mathbf{x} \in \mathbb{F}_q^n \;\middle|\; H\mathbf{x}^{\mathrm{T}} = \mathbf{0}\right\}. \tag{1.1}$$

Note that the rows of H will also be independent. In general, there are also several possible parity check matrices for $\mathcal{C}$. The next theorem gives one of them when $\mathcal{C}$ has a generator matrix in standard form. In this theorem A^{T} is the transpose of A.

Theorem 1.2.1 *If $G = [I_k \mid A]$ is a generator matrix for the $[n, k]$ code $\mathcal{C}$ in standard form, then $H = [-A^{\mathrm{T}} \mid I_{n-k}]$ is a parity check matrix for $\mathcal{C}$.*

Proof: We clearly have $HG^{\mathrm{T}} = -A^{\mathrm{T}} + A^{\mathrm{T}} = O$. Thus $\mathcal{C}$ is contained in the kernel of the linear transformation $\mathbf{x} \mapsto H\mathbf{x}^{\mathrm{T}}$. As H has rank $n - k$, this linear transformation has kernel of dimension k, which is also the dimension of $\mathcal{C}$. The result follows. □

Exercise 1 Prior to the statement of Theorem 1.2.1, it was noted that the rows of the $(n - k) \times n$ parity check matrix H satisfying (1.1) are independent. Why is that so? Hint: The map $\mathbf{x} \mapsto H\mathbf{x}^{\mathrm{T}}$ is a linear transformation from $\mathbb{F}_q^n$ to $\mathbb{F}_q^{n-k}$ with kernel $\mathcal{C}$. From linear algebra, what is the rank of H? ♦

Example 1.2.2 The simplest way to encode information in order to recover it in the presence of noise is to repeat each message symbol a fixed number of times. Suppose that our information is binary with symbols from the field $\mathbb{F}_2$, and we repeat each symbol n times. If for instance $n = 7$, then whenever we want to send a 0 we send 0000000, and whenever we want to send a 1 we send 1111111. If at most three errors are made in transmission and if we decode by "majority vote," then we can correctly determine the information symbol, 0 or 1. In general, our code $\mathcal{C}$ is the $[n, 1]$ binary linear code consisting of the two codewords $\mathbf{0} = 00 \cdots 0$ and $\mathbf{1} = 11 \cdots 1$ and is called the *binary repetition code* of length n. This code can correct up to $e = \lfloor (n - 1)/2 \rfloor$ errors: if at most e errors are made in a received vector, then the majority of coordinates will be correct, and hence the original sent codeword can be recovered. If more than e errors are made, these errors cannot be corrected. However, this code can detect $n - 1$ errors, as received vectors with between 1 and $n - 1$ errors will definitely not be codewords. A generator matrix for the repetition code is

$$G = [1 \mid 1 \quad \cdots \quad 1],$$

which is of course in standard form. The corresponding parity check matrix from Theorem 1.2.1 is

$$H = \left[\begin{array}{c|c} 1 & \\ 1 & \\ \vdots & I_{n-1} \\ 1 & \end{array}\right].$$

The first coordinate is an information set and the last $n - 1$ coordinates form a redundancy set. ■

Exercise 2 How many information sets are there for the $[n, 1]$ repetition code of Example 1.2.2? ♦

Example 1.2.3 The matrix $G = [I_4 \mid A]$, where

$$G = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right]$$

is a generator matrix in standard form for a $[7, 4]$ binary code that we denote by $\mathcal{H}_3$. By Theorem 1.2.1 a parity check matrix for $\mathcal{H}_3$ is

$$H = [A^{\mathrm{T}} \mid I_3] = \left[\begin{array}{cccc|ccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right].$$

This code is called the $[7, 4]$ *Hamming code*. ■

Exercise 3 Find at least four information sets in the $[7, 4]$ code $\mathcal{H}_3$ from Example 1.2.3. Find at least one set of four coordinates that do not form an information set. ♦

Often in this text we will refer to a *subcode* of a code $\mathcal{C}$. If $\mathcal{C}$ is not linear (or not known to be linear), a subcode of $\mathcal{C}$ is any subset of $\mathcal{C}$. If $\mathcal{C}$ is linear, a subcode will be a subset of $\mathcal{C}$ which must also be linear; in this case a subcode of $\mathcal{C}$ is a subspace of $\mathcal{C}$.

1.3 Dual codes

The generator matrix G of an $[n, k]$ code $\mathcal{C}$ is simply a matrix whose rows are independent and span the code. The rows of the parity check matrix H are independent; hence H is the generator matrix of some code, called the *dual* or *orthogonal* of $\mathcal{C}$ and denoted $\mathcal{C}^{\perp}$. Notice that $\mathcal{C}^{\perp}$ is an $[n, n-k]$ code. An alternate way to define the dual code is by using inner products.

Recall that the ordinary inner product of vectors $\mathbf{x} = x_1 \cdots x_n$, $\mathbf{y} = y_1 \cdots y_n$ in $\mathbb{F}_q^n$ is

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{n} x_i y_i.$$

Therefore from (1.1), we see that $\mathcal{C}^\perp$ can also be defined by

$$\mathcal{C}^\perp = \left\{ \mathbf{x} \in \mathbb{F}_q^n \mid \mathbf{x} \cdot \mathbf{c} = 0 \text{ for all } \mathbf{c} \in \mathcal{C} \right\}. \tag{1.2}$$

It is a simple exercise to show that if G and H are generator and parity check matrices, respectively, for $\mathcal{C}$, then H and G are generator and parity check matrices, respectively, for $\mathcal{C}^\perp$.

Exercise 4 Prove that if G and H are generator and parity check matrices, respectively, for $\mathcal{C}$, then H and G are generator and parity check matrices, respectively, for $\mathcal{C}^\perp$. ♦

Example 1.3.1 Generator and parity check matrices for the $[n, 1]$ repetition code $\mathcal{C}$ are given in Example 1.2.2. The dual code $\mathcal{C}^\perp$ is the $[n, n-1]$ code with generator matrix H and thus consists of all binary n-tuples $a_1 a_2 \cdots a_{n-1} b$, where $b = a_1 + a_2 + \cdots + a_{n-1}$ (addition in $\mathbb{F}_2$). The nth coordinate b is an overall parity check for the first $n-1$ coordinates chosen, therefore, so that the sum of all the coordinates equals 0. This makes it easy to see that G is indeed a parity check matrix for $\mathcal{C}^\perp$. The code $\mathcal{C}^\perp$ has the property that a single transmission error can be detected (since the sum of the coordinates will not be 0) but not corrected (since changing any one of the received coordinates will give a vector whose sum of coordinates will be 0). ■

A code $\mathcal{C}$ is *self-orthogonal* provided $\mathcal{C} \subseteq \mathcal{C}^\perp$ and *self-dual* provided $\mathcal{C} = \mathcal{C}^\perp$. The length n of a self-dual code is even and the dimension is $n/2$.

Exercise 5 Prove that a self-dual code has even length n and dimension $n/2$. ♦

Example 1.3.2 One generator matrix for the $[7, 4]$ Hamming code $\mathcal{H}_3$ is presented in Example 1.2.3. Let $\widehat{\mathcal{H}}_3$ be the code of length 8 and dimension 4 obtained from $\mathcal{H}_3$ by adding an overall parity check coordinate to each vector of G and thus to each codeword of $\mathcal{H}_3$. Then

$$\widehat{G} = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{array}\right]$$

is a generator matrix for $\widehat{\mathcal{H}}_3$. It is easy to verify that $\widehat{\mathcal{H}}_3$ is a self-dual code. ■

Example 1.3.3 The $[4, 2]$ ternary code $\mathcal{H}_{3,2}$, often called the *tetracode*, has generator matrix G, in standard form, given by

$$G = \left[\begin{array}{cc|cc} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & -1 \end{array}\right].$$

This code is also self-dual. ■

Exercise 6 Prove that $\widehat{\mathcal{H}}_3$ from Example 1.3.2 and $\mathcal{H}_{3,2}$ from Example 1.3.3 are self-dual codes. ♦

Exercise 7 Find all the information sets of the tetracode given in Example 1.3.3. ♦

When studying quaternary codes over the field $\mathbb{F}_4$ (Example 1.1.3), it is often useful to consider another inner product, called the *Hermitian inner product*, given by

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x} \cdot \overline{\mathbf{y}} = \sum_{i=1}^{n} x_i \overline{y_i},$$

where $\overline{}$, called *conjugation*, is given by $\overline{0} = 0$, $\overline{1} = 1$, and $\overline{\overline{\omega}} = \omega$. Using this inner product, we can define the *Hermitian dual* of a quaternary code $\mathcal{C}$ to be, analogous to (1.2),

$$\mathcal{C}^{\perp_H} = \left\{ \mathbf{x} \in \mathbb{F}_q^n \;\middle|\; \langle \mathbf{x}, \mathbf{c} \rangle = 0 \text{ for all } \mathbf{c} \in \mathcal{C} \right\}.$$

Define the *conjugate* of $\mathcal{C}$ to be

$$\overline{\mathcal{C}} = \{ \overline{\mathbf{c}} \mid \mathbf{c} \in \mathcal{C} \},$$

where $\overline{\mathbf{c}} = \overline{c_1}\,\overline{c_2} \cdots \overline{c_n}$ when $\mathbf{c} = c_1 c_2 \cdots c_n$. Notice that $\mathcal{C}^{\perp_H} = \overline{\mathcal{C}}^{\perp}$. We also have Hermitian self-orthogonality and Hermitian self-duality: namely, $\mathcal{C}$ is *Hermitian self-orthogonal* if $\mathcal{C} \subseteq \mathcal{C}^{\perp_H}$ and *Hermitian self-dual* if $\mathcal{C} = \mathcal{C}^{\perp_H}$.

Exercise 8 Prove that if $\mathcal{C}$ is a code over $\mathbb{F}_4$, then $\mathcal{C}^{\perp_H} = \overline{\mathcal{C}}^{\perp}$. ♦

Example 1.3.4 The $[6, 3]$ quaternary code $\mathcal{G}_6$ has generator matrix G_6 in standard form given by

$$G_6 = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & \omega & \omega \\ 0 & 1 & 0 & \omega & 1 & \omega \\ 0 & 0 & 1 & \omega & \omega & 1 \end{array}\right].$$

This code is often called the *hexacode*. It is Hermitian self-dual. ■

Exercise 9 Verify the following properties of the Hermitian inner product on $\mathbb{F}_4^n$:
(a) $\langle \mathbf{x}, \mathbf{x} \rangle \in \{0, 1\}$ for all $\mathbf{x} \in \mathbb{F}_4^n$.
(b) $\langle \mathbf{x}, \mathbf{y} + \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{x}, \mathbf{z} \rangle$ for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}_4^n$.
(c) $\langle \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{y}, \mathbf{z} \rangle$ for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}_4^n$.
(d) $\overline{\langle \mathbf{x}, \mathbf{y} \rangle} = \langle \mathbf{y}, \mathbf{x} \rangle$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{F}_4^n$.
(e) $\langle \alpha\mathbf{x}, \mathbf{y} \rangle = \alpha \langle \mathbf{x}, \mathbf{y} \rangle$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{F}_4^n$.
(f) $\langle \mathbf{x}, \alpha\mathbf{y} \rangle = \overline{\alpha} \langle \mathbf{x}, \mathbf{y} \rangle$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{F}_4^n$. ♦

Exercise 10 Prove that the hexacode $\mathcal{G}_6$ from Example 1.3.4 is Hermitian self-dual. ♦

1.4 Weights and distances

An important invariant of a code is the minimum distance between codewords. The (*Hamming*) *distance* $\mathrm{d}(\mathbf{x}, \mathbf{y})$ between two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$ is defined to be the number of coordinates in which $\mathbf{x}$ and $\mathbf{y}$ differ. The proofs of the following properties of distance are left as an exercise.

Theorem 1.4.1 *The distance function* $\mathrm{d}(\mathbf{x}, \mathbf{y})$ *satisfies the following four properties*:
(i) (*non-negativity*) $\mathrm{d}(\mathbf{x}, \mathbf{y}) \geq 0$ *for all* $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$.
(ii) $\mathrm{d}(\mathbf{x}, \mathbf{y}) = 0$ *if and only if* $\mathbf{x} = \mathbf{y}$.
(iii) (*symmetry*) $\mathrm{d}(\mathbf{x}, \mathbf{y}) = \mathrm{d}(\mathbf{y}, \mathbf{x})$ *for all* $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$.
(iv) (*triangle inequality*) $\mathrm{d}(\mathbf{x}, \mathbf{z}) \leq \mathrm{d}(\mathbf{x}, \mathbf{y}) + \mathrm{d}(\mathbf{y}, \mathbf{z})$ *for all* $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}_q^n$.

This theorem makes the distance function a metric on the vector space $\mathbb{F}_q^n$.

Exercise 11 Prove Theorem 1.4.1. ♦

The (*minimum*) *distance* of a code $\mathcal{C}$ is the smallest distance between distinct codewords and is important in determining the error-correcting capability of $\mathcal{C}$; as we see later, the higher the minimum distance, the more errors the code can correct. The (*Hamming*) *weight* $\mathrm{wt}(\mathbf{x})$ of a vector $\mathbf{x} \in \mathbb{F}_q^n$ is the number of nonzero coordinates in $\mathbf{x}$. The proof of the following relationship between distance and weight is also left as an exercise.

Theorem 1.4.2 *If* $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$, *then* $\mathrm{d}(\mathbf{x}, \mathbf{y}) = \mathrm{wt}(\mathbf{x} - \mathbf{y})$. *If* $\mathcal{C}$ *is a linear code, the minimum distance* d *is the same as the minimum weight of the nonzero codewords of* $\mathcal{C}$.

As a result of this theorem, for linear codes, the minimum distance is also called the *minimum weight* of the code. If the minimum weight d of an $[n, k]$ code is known, then we refer to the code as an $[n, k, d]$ code.

Exercise 12 Prove Theorem 1.4.2. ♦

When dealing with codes over $\mathbb{F}_2$, $\mathbb{F}_3$, or $\mathbb{F}_4$, there are some elementary results about codeword weights that prove to be useful. We collect them here and leave the proof to the reader.

Theorem 1.4.3 *The following hold*:
(i) *If* $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$, *then*

$$\mathrm{wt}(\mathbf{x} + \mathbf{y}) = \mathrm{wt}(\mathbf{x}) + \mathrm{wt}(\mathbf{y}) - 2\mathrm{wt}(\mathbf{x} \cap \mathbf{y}),$$

where $\mathbf{x} \cap \mathbf{y}$ *is the vector in* $\mathbb{F}_2^n$, *which has* 1*s precisely in those positions where both* $\mathbf{x}$ *and* $\mathbf{y}$ *have* 1*s*.
(ii) *If* $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$, *then* $\mathrm{wt}(\mathbf{x} \cap \mathbf{y}) \equiv \mathbf{x} \cdot \mathbf{y} \pmod 2$.
(iii) *If* $\mathbf{x} \in \mathbb{F}_2^n$, *then* $\mathrm{wt}(\mathbf{x}) \equiv \mathbf{x} \cdot \mathbf{x} \pmod 2$.
(iv) *If* $\mathbf{x} \in \mathbb{F}_3^n$, *then* $\mathrm{wt}(\mathbf{x}) \equiv \mathbf{x} \cdot \mathbf{x} \pmod 3$.
(v) *If* $\mathbf{x} \in \mathbb{F}_4^n$, *then* $\mathrm{wt}(\mathbf{x}) \equiv \langle \mathbf{x},\mathbf{x} \rangle \pmod 2$.

Exercise 13 Prove Theorem 1.4.3. ♦

Let A_i, also denoted $A_i(\mathcal{C})$, be the number of codewords of weight i in $\mathcal{C}$. The list A_i for $0 \leq i \leq n$ is called the *weight distribution* or *weight spectrum* of $\mathcal{C}$. A great deal of research is devoted to the computation of the weight distribution of specific codes or families of codes.

Example 1.4.4 Let $\mathcal{C}$ be the binary code with generator matrix

$$G = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

The weight distribution of $\mathcal{C}$ is $A_0 = A_6 = 1$ and $A_2 = A_4 = 3$. Notice that only the nonzero A_i are usually listed. ■

Exercise 14 Find the weight distribution of the ternary code with generator matrix

$$G = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Compare your result to Example 1.4.4. ♦

Certain elementary facts about the weight distribution are gathered in the following theorem. Deeper results on the weight distribution of codes will be presented in Chapter 7.

Theorem 1.4.5 *Let $\mathcal{C}$ be an $[n, k, d]$ code over $\mathbb{F}_q$. Then*:
(i) $A_0(\mathcal{C}) + A_1(\mathcal{C}) + \cdots + A_n(\mathcal{C}) = q^k$.
(ii) $A_0(\mathcal{C}) = 1$ *and* $A_1(\mathcal{C}) = A_2(\mathcal{C}) = \cdots = A_{d-1}(\mathcal{C}) = 0$.
(iii) *If $\mathcal{C}$ is a binary code containing the codeword* $\mathbf{1} = 11\cdots 1$, *then* $A_i(\mathcal{C}) = A_{n-i}(\mathcal{C})$ *for* $0 \leq i \leq n$.
(iv) *If $\mathcal{C}$ is a binary self-orthogonal code, then each codeword has even weight, and $\mathcal{C}^\perp$ contains the codeword* $\mathbf{1} = 11\cdots 1$.
(v) *If $\mathcal{C}$ is a ternary self-orthogonal code, then the weight of each codeword is divisible by three.*
(vi) *If $\mathcal{C}$ is a quaternary Hermitian self-orthogonal code, then the weight of each codeword is even.*

Exercise 15 Prove Theorem 1.4.5. ♦

Theorem 1.4.5(iv) states that all codewords in a binary self-orthogonal code $\mathcal{C}$ have even weight. If we look at the subset of codewords of $\mathcal{C}$ that have weights divisible by four, we surprisingly get a subcode of $\mathcal{C}$; that is, the subset of codewords of weights divisible by four form a subspace of $\mathcal{C}$. This is not necessarily the case for non-self-orthogonal codes.

Theorem 1.4.6 *Let $\mathcal{C}$ be an $[n, k]$ self-orthogonal binary code. Let $\mathcal{C}_0$ be the set of codewords in $\mathcal{C}$ whose weights are divisible by four. Then either*:
(i) $\mathcal{C} = \mathcal{C}_0$, *or*
(ii) *$\mathcal{C}_0$ is an $[n, k-1]$ subcode of $\mathcal{C}$ and $\mathcal{C} = \mathcal{C}_0 \cup \mathcal{C}_1$, where $\mathcal{C}_1 = \mathbf{x} + \mathcal{C}_0$ for any codeword $\mathbf{x}$ whose weight is even but not divisible by four. Furthermore $\mathcal{C}_1$ consists of all codewords of $\mathcal{C}$ whose weights are not divisible by four.*

Proof: By Theorem 1.4.5(iv) all codewords have even weight. Therefore either (i) holds or there exists a codeword $\mathbf{x}$ of even weight but not of weight a multiple of four. Assume the latter. Let $\mathbf{y}$ be another codeword whose weight is even but not a multiple of four. Then by Theorem 1.4.3(i), $\mathrm{wt}(\mathbf{x}+\mathbf{y}) = \mathrm{wt}(\mathbf{x}) + \mathrm{wt}(\mathbf{y}) - 2\mathrm{wt}(\mathbf{x} \cap \mathbf{y}) \equiv 2 + 2 - 2\mathrm{wt}(\mathbf{x} \cap \mathbf{y}) \pmod 4$. But by Theorem 1.4.3(ii), $\mathrm{wt}(\mathbf{x} \cap \mathbf{y}) \equiv \mathbf{x} \cdot \mathbf{y} \pmod 2$. Hence $\mathrm{wt}(\mathbf{x}+\mathbf{y})$ is divisible by four. Therefore $\mathbf{x}+\mathbf{y} \in \mathcal{C}_0$. This shows that $\mathbf{y} \in \mathbf{x} + \mathcal{C}_0$ and $\mathcal{C} = \mathcal{C}_0 \cup (\mathbf{x} + \mathcal{C}_0)$. That $\mathcal{C}_0$ is a subcode of $\mathcal{C}$ and that $\mathcal{C}_1 = \mathbf{x} + \mathcal{C}_0$ consists of all codewords of $\mathcal{C}$ whose weights are not divisible by four follow from a similar argument. □

There is an analogous result to Theorem 1.4.6 where you consider the subset of codewords of a binary code whose weights are even. In this case the self-orthogonality requirement is unnecessary; we leave its proof to the exercises.

Theorem 1.4.7 *Let $\mathcal{C}$ be an $[n, k]$ binary code. Let $\mathcal{C}_e$ be the set of codewords in $\mathcal{C}$ whose weights are even. Then either*:
(i) *$\mathcal{C} = \mathcal{C}_e$, or*
(ii) *$\mathcal{C}_e$ is an $[n, k-1]$ subcode of $\mathcal{C}$ and $\mathcal{C} = \mathcal{C}_e \cup \mathcal{C}_o$, where $\mathcal{C}_o = \mathbf{x} + \mathcal{C}_e$ for any codeword $\mathbf{x}$ whose weight is odd. Furthermore $\mathcal{C}_o$ consists of all codewords of $\mathcal{C}$ whose weights are odd.*

Exercise 16 Prove Theorem 1.4.7. ♦

Exercise 17 Let $\mathcal{C}$ be the $[6, 3]$ binary code with generator matrix

$$G = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

(a) Prove that $\mathcal{C}$ is not self-orthogonal.
(b) Find the weight distribution of $\mathcal{C}$.
(c) Show that the codewords whose weights are divisible by four do not form a subcode of $\mathcal{C}$. ♦

The next result gives a way to tell when Theorem 1.4.6(i) is satisfied.

Theorem 1.4.8 *Let $\mathcal{C}$ be a binary linear code.*
(i) *If $\mathcal{C}$ is self-orthogonal and has a generator matrix each of whose rows has weight divisible by four, then every codeword of $\mathcal{C}$ has weight divisible by four.*
(ii) *If every codeword of $\mathcal{C}$ has weight divisible by four, then $\mathcal{C}$ is self-orthogonal.*

Proof: For (i), let $\mathbf{x}$ and $\mathbf{y}$ be rows of the generator matrix. By Theorem 1.4.3(i), $\mathrm{wt}(\mathbf{x}+\mathbf{y}) = \mathrm{wt}(\mathbf{x}) + \mathrm{wt}(\mathbf{y}) - 2\mathrm{wt}(\mathbf{x} \cap \mathbf{y}) \equiv 0 + 0 - 2\mathrm{wt}(\mathbf{x} \cap \mathbf{y}) \equiv 0 \pmod 4$. Now proceed by induction as every codeword is a sum of rows of the generator matrix. For (ii), let $\mathbf{x}, \mathbf{y} \in \mathcal{C}$. By Theorem 1.4.3(i) and (ii), $2(\mathbf{x} \cdot \mathbf{y}) \equiv 2\mathrm{wt}(\mathbf{x} \cap \mathbf{y}) \equiv 2\mathrm{wt}(\mathbf{x} \cap \mathbf{y}) - \mathrm{wt}(\mathbf{x}) - \mathrm{wt}(\mathbf{y}) \equiv -\mathrm{wt}(\mathbf{x} + \mathbf{y}) \equiv 0 \pmod 4$. Thus $\mathbf{x} \cdot \mathbf{y} \equiv 0 \pmod 2$. □

It is natural to ask if Theorem 1.4.8(ii) can be generalized to codes whose codewords have weights that are divisible by numbers other than four. We say that a code $\mathcal{C}$ (over any field) is *divisible* provided all codewords have weights divisible by an integer $\Delta > 1$. The code is said to be *divisible by* Δ; Δ is called *a divisor* of $\mathcal{C}$, and the largest such divisor is called *the divisor* of $\mathcal{C}$. Thus Theorem 1.4.8(ii) says that binary codes divisible by $\Delta = 4$ are self-orthogonal. This is not true when considering binary codes divisible by $\Delta = 2$, as the next example illustrates. Binary codes divisible by $\Delta = 2$ are called *even*.

Example 1.4.9 The dual of the $[n, 1]$ binary repetition code $\mathcal{C}$ of Example 1.2.2 consists of all the even weight vectors of length n. (See also Example 1.3.1.) If $n > 2$, this code is not self-orthogonal. ■

When considering codes over $\mathbb{F}_3$ and $\mathbb{F}_4$, the divisible codes with divisors three and two, respectively, are self-orthogonal as the next theorem shows. This theorem includes the converse of Theorem 1.4.5(v) and (vi). Part (ii) is found in [217].

Theorem 1.4.10 *Let $\mathcal{C}$ be a code over $\mathbb{F}_q$, with $q = 3$ or 4.*

(i) *When $q = 3$, every codeword of $\mathcal{C}$ has weight divisible by three if and only if $\mathcal{C}$ is self-orthogonal.*

(ii) *When $q = 4$, every codeword of $\mathcal{C}$ has weight divisible by two if and only if $\mathcal{C}$ is Hermitian self-orthogonal.*

Proof: In (i), if $\mathcal{C}$ is self-orthogonal, the codewords have weights divisible by three by Theorem 1.4.5(v). For the converse let $\mathbf{x}, \mathbf{y} \in \mathcal{C}$. We need to show that $\mathbf{x} \cdot \mathbf{y} = 0$. We can view the codewords $\mathbf{x}$ and $\mathbf{y}$ having the following parameters:

$$\begin{array}{cccccc} \mathbf{x}: & \star & 0 & = & \neq & 0 \\ \mathbf{y}: & 0 & \star & = & \neq & 0 \\ & a & b & c & d & e \end{array}$$

where there are a coordinates where $\mathbf{x}$ is nonzero and $\mathbf{y}$ is zero, b coordinates where $\mathbf{y}$ is nonzero and $\mathbf{x}$ is zero, c coordinates where both agree and are nonzero, d coordinates when both disagree and are nonzero, and e coordinates where both are zero. So $\mathrm{wt}(\mathbf{x} + \mathbf{y}) = a + b + c$ and $\mathrm{wt}(\mathbf{x} - \mathbf{y}) = a + b + d$. But $\mathbf{x} \pm \mathbf{y} \in \mathcal{C}$ and hence $a + b + c \equiv a + b + d \equiv 0 \pmod 3$. In particular $c \equiv d \pmod 3$. Therefore $\mathbf{x} \cdot \mathbf{y} = c + 2d \equiv 0 \pmod 3$, proving (i).

In (ii), if $\mathcal{C}$ is Hermitian self-orthogonal, the codewords have even weights by Theorem 1.4.5(vi). For the converse let $\mathbf{x} \in \mathcal{C}$. If $\mathbf{x}$ has a 0s, b 1s, c ωs, and d $\overline{\omega}$s, then $b + c + d$ is even as $\mathrm{wt}(\mathbf{x}) = b + c + d$. However, $\langle \mathbf{x}, \mathbf{x} \rangle$ also equals $b + c + d$ (as an element of $\mathbb{F}_4$). Therefore $\langle \mathbf{x}, \mathbf{x} \rangle = 0$ for all $\mathbf{x} \in \mathcal{C}$. Now let $\mathbf{x}, \mathbf{y} \in \mathcal{C}$. So both $\mathbf{x} + \mathbf{y}$ and $\omega\mathbf{x} + \mathbf{y}$ are in $\mathcal{C}$. Using Exercise 9 we have $0 = \langle \mathbf{x} + \mathbf{y}, \mathbf{x} + \mathbf{y} \rangle = \langle \mathbf{x}, \mathbf{x} \rangle + \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{y}, \mathbf{x} \rangle + \langle \mathbf{y}, \mathbf{y} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{y}, \mathbf{x} \rangle$. Also $0 = \langle \omega\mathbf{x} + \mathbf{y}, \omega\mathbf{x} + \mathbf{y} \rangle = \langle \mathbf{x}, \mathbf{x} \rangle + \omega\langle \mathbf{x}, \mathbf{y} \rangle + \overline{\omega}\langle \mathbf{y}, \mathbf{x} \rangle + \langle \mathbf{y}, \mathbf{y} \rangle = \omega\langle \mathbf{x}, \mathbf{y} \rangle + \overline{\omega}\langle \mathbf{y}, \mathbf{x} \rangle$. Combining these $\langle \mathbf{x}, \mathbf{y} \rangle$ must be 0, proving (ii). □

The converse of Theorem 1.4.5(iv) is in general not true. The best that can be said in this case is contained in the following theorem, whose proof we leave as an exercise.

Theorem 1.4.11 *Let $\mathcal{C}$ be a binary code with a generator matrix each of whose rows has even weight. Then every codeword of $\mathcal{C}$ has even weight.*

Exercise 18 Prove Theorem 1.4.11. ♦

Binary codes for which all codewords have weight divisible by four are called *doubly-even*.[2] By Theorem 1.4.8, doubly-even codes are self-orthogonal. A self-orthogonal code must be even by Theorem 1.4.5(iv); one which is not doubly-even is called *singly-even*.

Exercise 19 Find the minimum weights and weight distributions of the codes $\mathcal{H}_3$ in Example 1.2.3, $\mathcal{H}_3^{\perp}$, $\widehat{\mathcal{H}}_3$ in Example 1.3.2, the tetracode in Example 1.3.3, and the hexacode in Example 1.3.4. Which of the binary codes listed are self-orthogonal? Which are doubly-even? Which are singly-even? ♦

There is a generalization of the concepts of even and odd weight binary vectors to vectors over arbitrary fields, which is useful in the study of many types of codes. A vector $\mathbf{x} = x_1x_2\cdots x_n$ in $\mathbb{F}_q^n$ is *even-like* provided that

$$\sum_{i=1}^{n} x_i = 0$$

and is *odd-like* otherwise. A binary vector is even-like if and only if it has even weight; so the concept of even-like vectors is indeed a generalization of even weight binary vectors. The even-like vectors in a code form a subcode of a code over $\mathbb{F}_q$ as did the even weight vectors in a binary code. Except in the binary case, even-like vectors need not have even weight. The vectors $(1, 1, 1)$ in $\mathbb{F}_3^3$ and $(1, \omega, \overline{\omega})$ in $\mathbb{F}_4^3$ are examples. We say that a code is *even-like* if it has only even-like codewords; a code is *odd-like* if it is not even-like.

Theorem 1.4.12 *Let $\mathcal{C}$ be an $[n, k]$ code over $\mathbb{F}_q$. Let $\mathcal{C}_e$ be the set of even-like codewords in $\mathcal{C}$. Then either*:
(i) $\mathcal{C} = \mathcal{C}_e$, *or*
(ii) $\mathcal{C}_e$ *is an* $[n, k-1]$ *subcode of* $\mathcal{C}$.

Exercise 20 Prove Theorem 1.4.12. ♦

There is an elementary relationship between the weight of a codeword and a parity check matrix for a linear code. This is presented in the following theorem whose proof is left as an exercise.

Theorem 1.4.13 *Let $\mathcal{C}$ be a linear code with parity check matrix H. If $\mathbf{c} \in \mathcal{C}$, the columns of H corresponding to the nonzero coordinates of $\mathbf{c}$ are linearly dependent. Conversely, if a linear dependence relation with nonzero coefficients exists among w columns of H, then there is a codeword in $\mathcal{C}$ of weight w whose nonzero coordinates correspond to these columns.*

One way to find the minimum weight d of a linear code is to examine all the nonzero codewords. The following corollary shows how to use the parity check matrix to find d.

[2] Some authors reserve the term "doubly-even" for self-dual codes for which all codewords have weight divisible by four.

Corollary 1.4.14 *A linear code has minimum weight d if and only if its parity check matrix has a set of d linearly dependent columns but no set of $d - 1$ linearly dependent columns.*

Exercise 21 Prove Theorem 1.4.13 and Corollary 1.4.14. ♦

The minimum weight is also characterized in the following theorem.

Theorem 1.4.15 *If $\mathcal{C}$ is an $[n, k, d]$ code, then every $n - d + 1$ coordinate position contains an information set. Furthermore, d is the largest number with this property.*

Proof: Let G be a generator matrix for $\mathcal{C}$, and consider any set X of s coordinate positions. To make the argument easier, we assume X is the set of the last s positions. (After we develop the notion of equivalent codes, the reader will see that this argument is in fact general.) Suppose X does not contain an information set. Let $G = [A \mid B]$, where A is $k \times (n - s)$ and B is $k \times s$. Then the column rank of B, and hence the row rank of B, is less than k. Hence there exists a nontrivial linear combination of the rows of B which equals $\mathbf{0}$, and hence a codeword $\mathbf{c}$ which is $\mathbf{0}$ in the last s positions. Since the rows of G are linearly independent, $\mathbf{c} \neq \mathbf{0}$ and hence $d \leq n - s$, equivalently, $s \leq n - d$. The theorem now follows. □

Exercise 22 Find the number of information sets for the $[7, 4]$ Hamming code $\mathcal{H}_3$ given in Example 1.2.3. Do the same for the extended Hamming code $\widehat{\mathcal{H}}_3$ from Example 1.3.2. ♦

1.5 New codes from old

As we will see throughout this book, many interesting and important codes will arise by modifying or combining existing codes. We will discuss five ways to do this.

1.5.1 Puncturing codes

Let $\mathcal{C}$ be an $[n, k, d]$ code over $\mathbb{F}_q$. We can *puncture* $\mathcal{C}$ by deleting the same coordinate i in each codeword. The resulting code is still linear, a fact that we leave as an exercise; its length is $n - 1$, and we often denote the punctured code by $\mathcal{C}^*$. If G is a generator matrix for $\mathcal{C}$, then a generator matrix for $\mathcal{C}^*$ is obtained from G by deleting column i (and omitting a zero or duplicate row that may occur). What are the dimension and minimum weight of $\mathcal{C}^*$? Because $\mathcal{C}$ contains q^k codewords, the only way that $\mathcal{C}^*$ could contain fewer codewords is if two codewords of $\mathcal{C}$ agree in all but coordinate i. In that case $\mathcal{C}$ has minimum distance $d = 1$ and a codeword of weight 1 whose nonzero entry is in coordinate i. The minimum distance decreases by 1 only if a minimum weight codeword of $\mathcal{C}$ has a nonzero ith coordinate. Summarizing, we have the following theorem.

Theorem 1.5.1 *Let $\mathcal{C}$ be an $[n, k, d]$ code over $\mathbb{F}_q$, and let $\mathcal{C}^*$ be the code $\mathcal{C}$ punctured on the ith coordinate.*

(i) *If* $d > 1$, $\mathcal{C}^*$ *is an* $[n-1, k, d^*]$ *code where* $d^* = d - 1$ *if* $\mathcal{C}$ *has a minimum weight codeword with a nonzero* i*th coordinate and* $d^* = d$ *otherwise.*
(ii) *When* $d = 1$, $\mathcal{C}^*$ *is an* $[n-1, k, 1]$ *code if* $\mathcal{C}$ *has no codeword of weight* 1 *whose nonzero entry is in coordinate* i*; otherwise, if* $k > 1$, $\mathcal{C}^*$ *is an* $[n-1, k-1, d^*]$ *code with* $d^* \geq 1$.

Exercise 23 Prove directly from the definition that a punctured linear code is also linear. ♦

Example 1.5.2 Let $\mathcal{C}$ be the $[5, 2, 2]$ binary code with generator matrix

$$G = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

Let $\mathcal{C}_1^*$ and $\mathcal{C}_5^*$ be the code $\mathcal{C}$ punctured on coordinates 1 and 5, respectively. They have generator matrices

$$G_1^* = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \quad \text{and} \quad G_5^* = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

So $\mathcal{C}_1^*$ is a $[4, 2, 1]$ code, while $\mathcal{C}_5^*$ is a $[4, 2, 2]$ code. ■

Example 1.5.3 Let $\mathcal{D}$ be the $[4, 2, 1]$ binary code with generator matrix

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}.$$

Let $\mathcal{D}_1^*$ and $\mathcal{D}_4^*$ be the code $\mathcal{D}$ punctured on coordinates 1 and 4, respectively. They have generator matrices

$$D_1^* = [1 \quad 1 \quad 1] \quad \text{and} \quad D_4^* = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

So $\mathcal{D}_1^*$ is a $[3, 1, 3]$ code and $\mathcal{D}_4^*$ is a $[3, 2, 1]$ code. ■

Notice that the code $\mathcal{D}$ of Example 1.5.3 is the code $\mathcal{C}_1^*$ of Example 1.5.2. Obviously $\mathcal{D}_4^*$ could have been obtained from $\mathcal{C}$ directly by puncturing on coordinates $\{1, 5\}$. In general a code $\mathcal{C}$ can be punctured on the coordinate set T by deleting components indexed by the set T in all codewords of $\mathcal{C}$. If T has size t, the resulting code, which we will often denote $\mathcal{C}^T$, is an $[n-t, k^*, d^*]$ code with $k^* \geq k - t$ and $d^* \geq d - t$ by Theorem 1.5.1 and induction.

1.5.2 Extending codes

We can create longer codes by adding a coordinate. There are many possible ways to extend a code but the most common is to choose the extension so that the new code has only even-like vectors (as defined in Section 1.4). If $\mathcal{C}$ is an $[n, k, d]$ code over $\mathbb{F}_q$, define the *extended* code $\widehat{\mathcal{C}}$ to be the code

$$\widehat{\mathcal{C}} = \left\{ x_1 x_2 \cdots x_{n+1} \in \mathbb{F}_q^{n+1} \;\middle|\; x_1 x_2 \cdots x_n \in \mathcal{C} \text{ with } x_1 + x_2 + \cdots + x_{n+1} = 0 \right\}.$$

We leave it as an exercise to show that $\widehat{\mathcal{C}}$ is linear. In fact $\widehat{\mathcal{C}}$ is an $[n+1, k, \widehat{d}]$ code, where $\widehat{d} = d$ or $d+1$. Let G and H be generator and parity check matrices, respectively, for $\mathcal{C}$. Then a generator matrix $\widehat{G}$ for $\widehat{\mathcal{C}}$ can be obtained from G by adding an extra column to G so that the sum of the coordinates of each row of $\widehat{G}$ is 0. A parity check matrix for $\widehat{\mathcal{C}}$ is the matrix

$$\widehat{H} = \left[\begin{array}{ccc|c} 1 & \cdots & 1 & 1 \\ \hline & & & 0 \\ & H & & \vdots \\ & & & 0 \end{array}\right]. \tag{1.3}$$

This construction is also referred to as *adding an overall parity check*. The [8, 4, 4] binary code $\widehat{\mathcal{H}}_3$ in Example 1.3.2 obtained from the [7, 4, 3] Hamming code $\mathcal{H}_3$ by adding an overall parity check is called the *extended Hamming code*.

Exercise 24 Prove directly from the definition that an extended linear code is also linear. ♦

Exercise 25 Suppose we extend the $[n, k]$ linear code $\mathcal{C}$ over the field $\mathbb{F}_q$ to the code $\widetilde{\mathcal{C}}$ where

$$\widetilde{\mathcal{C}} = \left\{x_1x_2\cdots x_{n+1} \in \mathbb{F}_q^{n+1} \mid x_1x_2\cdots x_n \in \mathcal{C} \text{ with } x_1^2 + x_2^2 + \cdots + x_{n+1}^2 = 0\right\}.$$

Under what conditions is $\widetilde{\mathcal{C}}$ linear? ♦

Exercise 26 Prove that $\widehat{H}$ in (1.3) is the parity check matrix for an extended code $\widehat{\mathcal{C}}$, where $\mathcal{C}$ has parity check matrix H. ♦

If $\mathcal{C}$ is an $[n, k, d]$ binary code, then the extended code $\widehat{\mathcal{C}}$ contains only even weight vectors and is an $[n+1, k, \widehat{d}]$ code, where $\widehat{d}$ equals d if d is even and equals $d+1$ if d is odd. This is consistent with the results obtained by extending $\mathcal{H}_3$. In the nonbinary case, however, whether or not $\widehat{d}$ is d or $d+1$ is not so straightforward. For an $[n, k, d]$ code $\mathcal{C}$ over $\mathbb{F}_q$, call the minimum weight of the even-like codewords, respectively the odd-like codewords, the *minimum even-like weight*, respectively the *minimum odd-like weight*, of the code. Denote the minimum even-like weight by d_e and the minimum odd-like weight by d_o. So $d = \min\{d_e, d_o\}$. If $d_e \le d_o$, then $\widehat{\mathcal{C}}$ has minimum weight $\widehat{d} = d_e$. If $d_o < d_e$, then $\widehat{d} = d_o + 1$.

Example 1.5.4 Recall that the tetracode $\mathcal{H}_{3,2}$ from Example 1.3.3 is a [4, 2, 3] code over $\mathbb{F}_3$ with generator matrix G and parity check matrix H given by

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & -1 \end{bmatrix} \quad \text{and} \quad H = \begin{bmatrix} -1 & -1 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{bmatrix}.$$

The codeword (1, 0, 1, 1) extends to (1, 0, 1, 1, 0) and the codeword (0, 1, 1, −1) extends to (0, 1, 1, −1, −1). Hence $d = d_e = d_o = 3$ and $\widehat{d} = 3$. The generator and parity check matrices for $\widehat{\mathcal{H}}_{3,2}$ are

$$\widehat{G} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & -1 & -1 \end{bmatrix} \quad \text{and} \quad \widehat{H} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

■

If we extend a code and then puncture the new coordinate, we obtain the original code. However, performing the operations in the other order will in general result in a different code.

Example 1.5.5 If we puncture the binary code $\mathcal{C}$ with generator matrix

$$G = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

on its last coordinate and then extend (on the right), the resulting code has generator matrix

$$G = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

■

In this example, our last step was to extend a binary code with only even weight vectors. The extended coordinate was always 0. In general, that is precisely what happens when you extend a code that has only even-like codewords.

Exercise 27 Do the following.

(a) Let $\mathcal{C} = \mathcal{H}_{3,2}$ be the $[4, 2, 3]$ tetracode over $\mathbb{F}_3$ defined in Example 1.3.3 with generator matrix

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & -1 \end{bmatrix}.$$

Give the generator matrix of the code obtained from $\mathcal{C}$ by puncturing on the right-most coordinate and then extending on the right. Also determine the minimum weight of the resulting code.

(b) Let $\mathcal{C}$ be a code over $\mathbb{F}_q$. Let $\mathcal{C}_1$ be the code obtained from $\mathcal{C}$ by puncturing on the right-most coordinate and then extending this punctured code on the right. Prove that $\mathcal{C} = \mathcal{C}_1$ if and only if $\mathcal{C}$ is an even-like code.

(c) With $\mathcal{C}_1$ defined as in (b), prove that if $\mathcal{C}$ is self-orthogonal and contains the all-one codeword $\mathbf{1}$, then $\mathcal{C} = \mathcal{C}_1$.

(d) With $\mathcal{C}_1$ defined as in (b), prove that $\mathcal{C} = \mathcal{C}_1$ if and only if the all-one vector $\mathbf{1}$ is in $\mathcal{C}^\perp$. ♦

1.5.3 Shortening codes

Let $\mathcal{C}$ be an $[n, k, d]$ code over $\mathbb{F}_q$ and let T be any set of t coordinates. Consider the set $\mathcal{C}(T)$ of codewords which are $\mathbf{0}$ on T; this set is a subcode of $\mathcal{C}$. Puncturing $\mathcal{C}(T)$ on T gives a code over $\mathbb{F}_q$ of length $n - t$ called the code *shortened* on T and denoted $\mathcal{C}_T$.

Example 1.5.6 Let $\mathcal{C}$ be the [6, 3, 2] binary code with generator matrix

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

$\mathcal{C}^\perp$ is also a [6, 3, 2] code with generator matrix

$$G^\perp = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

If the coordinates are labeled $1, 2, \ldots, 6$, let $T = \{5, 6\}$. Generator matrices for the shortened code $\mathcal{C}_T$ and punctured code $\mathcal{C}^T$ are

$$G_T = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad \text{and} \quad G^T = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Shortening and puncturing the dual code gives the codes $(\mathcal{C}^\perp)_T$ and $(\mathcal{C}^\perp)^T$, which have generator matrices

$$(G^\perp)_T = [1 \quad 1 \quad 1 \quad 1] \quad \text{and} \quad (G^\perp)^T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

From the generator matrices G_T and G^T, we find that the duals of $\mathcal{C}_T$ and $\mathcal{C}^T$ have generator matrices

$$(G_T)^\perp = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad (G^T)^\perp = [1 \quad 1 \quad 1 \quad 1].$$

Notice that these matrices show that $(\mathcal{C}^\perp)_T = (\mathcal{C}^T)^\perp$ and $(\mathcal{C}^\perp)^T = (\mathcal{C}_T)^\perp$. ■

The conclusions observed in the previous example hold in general.

Theorem 1.5.7 *Let $\mathcal{C}$ be an $[n, k, d]$ code over $\mathbb{F}_q$. Let T be a set of t coordinates. Then*:

(i) $(\mathcal{C}^\perp)_T = (\mathcal{C}^T)^\perp$ *and* $(\mathcal{C}^\perp)^T = (\mathcal{C}_T)^\perp$, *and*

(ii) *if $t < d$, then $\mathcal{C}^T$ and $(\mathcal{C}^\perp)_T$ have dimensions k and $n - t - k$, respectively*;

(iii) *if $t = d$ and T is the set of coordinates where a minimum weight codeword is nonzero, then $\mathcal{C}^T$ and $(\mathcal{C}^\perp)_T$ have dimensions $k - 1$ and $n - d - k + 1$, respectively.*

Proof: Let $\mathbf{c}$ be a codeword of $\mathcal{C}^\perp$ which is $\mathbf{0}$ on T and $\mathbf{c}^*$ the codeword with the coordinates in T removed. So $\mathbf{c}^* \in (\mathcal{C}^\perp)_T$. If $\mathbf{x} \in \mathcal{C}$, then $0 = \mathbf{x} \cdot \mathbf{c} = \mathbf{x}^* \cdot \mathbf{c}^*$, where $\mathbf{x}^*$ is the codeword $\mathbf{x}$ punctured on T. Thus $(\mathcal{C}^\perp)_T \subseteq (\mathcal{C}^T)^\perp$. Any vector $\mathbf{c} \in (\mathcal{C}^T)^\perp$ can be extended to a vector $\widehat{\mathbf{c}}$ by inserting 0s in the positions of T. If $\mathbf{x} \in \mathcal{C}$, puncture $\mathbf{x}$ on T to obtain $\mathbf{x}^*$. As $0 = \mathbf{x}^* \cdot \mathbf{c} = \mathbf{x} \cdot \widehat{\mathbf{c}}$, $\mathbf{c} \in (\mathcal{C}^\perp)_T$. Thus $(\mathcal{C}^\perp)_T = (\mathcal{C}^T)^\perp$. Replacing $\mathcal{C}$ by $\mathcal{C}^\perp$ gives $(\mathcal{C}^\perp)^T = (\mathcal{C}_T)^\perp$, completing (i).

Assume $t < d$. Then $n - d + 1 \le n - t$, implying any $n - t$ coordinates of $\mathcal{C}$ contain an information set by Theorem 1.4.15. Therefore $\mathcal{C}^T$ must be k-dimensional and hence $(\mathcal{C}^\perp)_T = (\mathcal{C}^T)^\perp$ has dimension $n - t - k$ by (i); this proves (ii).

As in (ii), (iii) is completed if we show that $\mathcal{C}^T$ has dimension $k-1$. If $S \subset T$ with S of size $d-1$, $\mathcal{C}^S$ has dimension k by part (ii). Clearly $\mathcal{C}^S$ has minimum distance 1 and $\mathcal{C}^T$ is obtained by puncturing $\mathcal{C}^S$ on the nonzero coordinate of a weight 1 codeword in $\mathcal{C}^S$. By Theorem 1.5.1(ii) $\mathcal{C}^T$ has dimension $k-1$. □

Exercise 28 Let $\mathcal{C}$ be the binary repetition code of length n as described in Example 1.2.2. Describe $(\mathcal{C}^\perp)_T$ and $(\mathcal{C}_T)^\perp$ for any T. ♦

Exercise 29 Let $\mathcal{C}$ be the code of length 6 in Example 1.4.4. Give generator matrices for $(\mathcal{C}^\perp)_T$ and $(\mathcal{C}_T)^\perp$ when $T = \{1, 2\}$ and $T = \{1, 3\}$. ♦

1.5.4 Direct sums

For $i \in \{1, 2\}$ let $\mathcal{C}_i$ be an $[n_i, k_i, d_i]$ code, both over the same finite field $\mathbb{F}_q$. Then their *direct sum* is the $[n_1 + n_2, k_1 + k_2, \min\{d_1, d_2\}]$ code

$$\mathcal{C}_1 \oplus \mathcal{C}_2 = \{(\mathbf{c}_1, \mathbf{c}_2) \mid \mathbf{c}_1 \in \mathcal{C}_1, \mathbf{c}_2 \in \mathcal{C}_2\}.$$

If $\mathcal{C}_i$ has generator matrix G_i and parity check matrix H_i, then

$$G_1 \oplus G_2 = \begin{bmatrix} G_1 & O \\ O & G_2 \end{bmatrix} \quad \text{and} \quad H_1 \oplus H_2 = \begin{bmatrix} H_1 & O \\ O & H_2 \end{bmatrix} \tag{1.4}$$

are a generator matrix and parity check matrix for $\mathcal{C}_1 \oplus \mathcal{C}_2$.

Exercise 30 Let $\mathcal{C}_i$ have generator matrix G_i and parity check matrix H_i for $i \in \{1, 2\}$. Prove that the generator and parity check matrices for $\mathcal{C}_1 \oplus \mathcal{C}_2$ are as given in (1.4). ♦

Exercise 31 Let $\mathcal{C}$ be the binary code with generator matrix

$$G = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Give another generator matrix for $\mathcal{C}$ that shows that $\mathcal{C}$ is a direct sum of two binary codes. ♦

Example 1.5.8 The $[6, 3, 2]$ binary code $\mathcal{C}$ of Example 1.4.4 is the direct sum $\mathcal{D} \oplus \mathcal{D} \oplus \mathcal{D}$ of the $[2, 1, 2]$ code $\mathcal{D} = \{00, 11\}$. ■

Since the minimum distance of the direct sum of two codes does not exceed the minimum distance of either of the codes, the direct sum of two codes is generally of little use in applications and is primarily of theoretical interest.

1.5.5 The $(\mathbf{u} \mid \mathbf{u} + \mathbf{v})$ construction

Two codes of the same length can be combined to form a third code of twice the length in a way similar to the direct sum construction. Let $\mathcal{C}_i$ be an $[n, k_i, d_i]$ code for $i \in \{1, 2\}$, both over the same finite field $\mathbb{F}_q$. The $(\mathbf{u} \mid \mathbf{u}+\mathbf{v})$ *construction* produces the $[2n, k_1 + k_2, \min\{2d_1, d_2\}]$ code

$$\mathcal{C} = \{(\mathbf{u}, \mathbf{u}+\mathbf{v}) \mid \mathbf{u} \in \mathcal{C}_1, \mathbf{v} \in \mathcal{C}_2\}.$$

If $\mathcal{C}_i$ has generator matrix G_i and parity check matrix H_i, then generator and parity check matrices for $\mathcal{C}$ are

$$\begin{bmatrix} G_1 & G_1 \\ O & G_2 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} H_1 & O \\ -H_2 & H_2 \end{bmatrix}. \tag{1.5}$$

Exercise 32 Prove that generator and parity check matrices for the code obtained in the $(\mathbf{u} \mid \mathbf{u}+\mathbf{v})$ construction from the codes $\mathcal{C}_i$ are as given in (1.5). ♦

Example 1.5.9 Consider the [8, 4, 4] binary code $\mathcal{C}$ with generator matrix

$$G = \left[\begin{array}{cccc|cccc} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right].$$

Then $\mathcal{C}$ can be produced from the [4, 3, 2] code $\mathcal{C}_1$ and the [4, 1, 4] code $\mathcal{C}_2$ with generator matrices

$$G_1 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \text{and} \quad G_2 = [1 \quad 1 \quad 1 \quad 1],$$

respectively, using the $(\mathbf{u} \mid \mathbf{u}+\mathbf{v})$ construction. Notice that the code $\mathcal{C}_1$ is also constructed using the $(\mathbf{u} \mid \mathbf{u}+\mathbf{v})$ construction from the [2, 2, 1] code $\mathcal{C}_3$ and the [2, 1, 2] code $\mathcal{C}_4$ with generator matrices

$$G_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad G_4 = [1 \quad 1],$$

respectively. ■

Unlike the direct sum construction of the previous section, the $(\mathbf{u} \mid \mathbf{u}+\mathbf{v})$ construction can produce codes that are important for reasons other than theoretical. For example, the family of Reed–Muller codes can be constructed in this manner as we see in Section 1.10. The code in the previous example is one of these codes.

Exercise 33 Prove that the $(\mathbf{u} \mid \mathbf{u}+\mathbf{v})$ construction using $[n, k_i, d_i]$ codes $\mathcal{C}_i$ produces a code of dimension $k = k_1 + k_2$ and minimum weight $d = \min\{2d_1, d_2\}$. ♦

1.6 Permutation equivalent codes

In this section and the next, we ask when two codes are "essentially the same." We term this concept "equivalence." Often we are interested in properties of codes, such as weight distribution, which remain unchanged when passing from one code to another that is essentially the same. Here we focus on the simplest form of equivalence, called permutation equivalence, and generalize this concept in the next section.

One way to view codes as "essentially the same" is to consider them "the same" if they are isomorphic as vector spaces. However, in that case the concept of weight, which we will see is crucial to the study and use of codes, is lost: codewords of one weight may be sent to codewords of a different weight by the isomorphism. A theorem of MacWilliams [212], which we will examine in Section 7.9, states that a vector space isomorphism of two binary codes of length n that preserves the weight of codewords (that is, send codewords of one weight to codewords of the same weight) can be extended to an isomorphism of $\mathbb{F}_2^n$ that is a permutation of coordinates. Clearly any permutation of coordinates that sends one code to another preserves the weight of codewords, regardless of the field. This leads to the following natural definition of permutation equivalent codes.

Two linear codes $\mathcal{C}_1$ and $\mathcal{C}_2$ are *permutation equivalent* provided there is a permutation of coordinates which sends $\mathcal{C}_1$ to $\mathcal{C}_2$. This permutation can be described using a *permutation matrix*, which is a square matrix with exactly one 1 in each row and column and 0s elsewhere. Thus $\mathcal{C}_1$ and $\mathcal{C}_2$ are permutation equivalent provided there is a permutation matrix P such that G_1 is a generator matrix of $\mathcal{C}_1$ if and only if G_1P is a generator matrix of $\mathcal{C}_2$. The effect of applying P to a generator matrix is to rearrange the columns of the generator matrix. If P is a permutation sending $\mathcal{C}_1$ to $\mathcal{C}_2$, we will write $\mathcal{C}_1P = \mathcal{C}_2$, where $\mathcal{C}_1P = \{\mathbf{y} \mid \mathbf{y} = \mathbf{x}P \text{ for } \mathbf{x} \in \mathcal{C}_1\}$.

Exercise 34 Prove that if G_1 and G_2 are generator matrices for a code $\mathcal{C}$ of length n and P is an $n \times n$ permutation matrix, then G_1P and G_2P are generator matrices for $\mathcal{C}P$. ♦

Exercise 35 Suppose $\mathcal{C}_1$ and $\mathcal{C}_2$ are permutation equivalent codes where $\mathcal{C}_1P = \mathcal{C}_2$ for some permutation matrix P. Prove that:
(a) $\mathcal{C}_1^\perp P = \mathcal{C}_2^\perp$, and
(b) if $\mathcal{C}_1$ is self-dual, so is $\mathcal{C}_2$. ♦

Example 1.6.1 Let $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$ be binary codes with generator matrices

$$G_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \quad G_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \text{and}$$

$$G_3 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

respectively. All three codes have weight distribution $A_0 = A_6 = 1$ and $A_2 = A_4 = 3$. (See Example 1.4.4 and Exercise 17.) The permutation switching columns 2 and 6 sends G_1 to G_2, showing that $\mathcal{C}_1$ and $\mathcal{C}_2$ are permutation equivalent. Both $\mathcal{C}_1$ and $\mathcal{C}_2$ are self-dual, consistent with (a) of Exercise 35. $\mathcal{C}_3$ is not self-dual. Therefore $\mathcal{C}_1$ and $\mathcal{C}_3$ are not permutation equivalent by part (b) of Exercise 35. ■