Chapter 5

Linear Coding Theory

5.1 Introduction

The purpose of this chapter is to give an introduction to linear coding theory. This is a topic that is not usually treated in linear algebra, but perhaps it should be. The point is that coding theory is based on elemnetary linear algebra, but it uses the finite fields $\mathbb{F}_p$ instead of the reals $\mathbb{R}$. Coding theory is an extremely important topic because without it, we wouldn't have PCs, modems, compact discs, DVDs and many other of the daily necessities.

Before starting, let's give a little history one of the contributions coding theory has made. In the 1960's and 70's, NASA launched several of the Mariner space probes in order to gather information about our planetary system. One of main the problems the NASA engineers faced was how to send the data which was gathered back to earth. Data which has to be sent electronically is encoded as binary strings, that is, strings of 0's and 1's. Since the space probes carried only a tiny, weak transmitter, there was a high probability that the transmitted data could be scrampled or entirely lost, due to the fact that there is a lot of radiation in space capable of disrupting commumications. Solar flares, for example, routinely make communications even here on earth an impossibility.

To compensate for the possible damage to the transmitted binary strings, NASA used what are called error-correcting codes. In an error-correcting code, only certain of the strings 0's and 1's, called codewords, are used, so that for any received string which isn't a codeword, there may be a good choice as to which codeword should be substituted to restore the integrity of the transmission. For example, the Mariner probe to Venus used an error-correcting code consisting of 64 codewords. Each codeword was a string of

32 0's and 1's (thus the codewords are elements of $(\mathbb{F}_2)^{32}$). This code had the remarkably good property that it was able to correct an errant reception with to 7 errors. In other words, almost 25% of the digits could be off and the correct codeword would still be deducible.

For the reader who wishes to pursue coding theory more deeply, there are several elementary texts, such as *Introduction to Coding Theory* by R. Hill and *Introduction to the Theory of Error-Correcting Codes* by V. Pless. A more advanced book is *Applied Abstract Algebra* by R. Lidl and G. Pilz. Though more demanding, this book discusses many interesting applications of linear algebra besides coding theory. The web is also an excellent source of information. Just type your search topic into www.google.com.

5.2 Linear Codes

5.2.1 The Notion of a Code

The purpose of this section is to introduce the notion of a code. Recall that $V(n,p)$ denotes $\mathbb{F}^n$, where $\mathbb{F}$ is the prime field $\mathbb{F}_p$.

Definition 5.1. A *p-ary code of length n* is defined to be a subset of C of $V(n,p)$. The elements of C are called *codewords*. We will denote the number of elements of C by $|C|$.

Since $|V(n,p) = p^n$, every code $C \subset V(n,p)$ is finite.

Proposition 5.1. *The number of codes $C \subset V(n,p)$ is 2^{p^n}.*

Proof. The number of subsets of a set with k elements is 2^k, while $|V(np)| = p^n$. □

Definition 5.2. A linear subspace $C \subset V(n,p)$ is called a *linear code*. (or more precisely, a *p-ary linear code of length n*).

Thus a code $C \subset V(n,p)$ with the property that the sum of any two codewords is a codeword, which also contains the null word (i.e. zero vector) is a p-ary linear code. (Note that when the field is $\mathbb{F}_p$, a subset containing the null word which is closed under addition is a subspace.)

An important advantage of linear codes is that a linear code is determined by giving a set of codewords which span it.

Definition 5.3. If $C \subset V(n,p)$ is linear, then any set of codewords which gives a basis of C is called a set of *basic codewords*. If $\dim C = k$, we call C a *p-ary $[n,k]$-code*.

Proposition 5.2. *If C is a p-ary $[n,k]$-code, then $|C| = p^k$.*

Proof. This is a special case of Proposition 4.7, but let's repeat the proof anyway. Since $\dim C = k$, C has a basis consisting of k codewords, say $\mathbf{c}_1, \ldots \mathbf{c}_k$.

Now every codeword can be expressed in exactly one way as a linear combination

$$a_1\mathbf{c}_1 + a_2\mathbf{c}_2 + \cdots a_k\mathbf{c}_k,$$

where $a_1, a_2, \ldots, a_k$ vary over all elements of $\mathbb{F}_p$. Hence there are at most p^k possible linear combinations. But different linear combinations give different vectors, so in fact $|C| = p^k$. □

The most frequently used codes are *binary codes*, that is codes where $\mathbb{F} = \mathbb{F}_2$, so we will concentrate on these. The elements of $V(n,2)$ will be represented simply as strings of n 0's and 1's. We will frequently refer to these as *n-bit strings*. For example, the two-bit strings are 00, 01, 10, and 11.

Example 5.1. The equation $x_1+x_2+x_3+x_4 = 0$ defines a 4-bit linear code of dimension 3. Hence there are 8 codewords. Rewriting this equation as $x_1+x_2+x_3 = x_4$, we see that x_4 can be viewed as a check digit for x_1, x_2, x_3. In this code, the codewords are the 4 bit strings with an even number of 1's. A particular set of basic codewords is $\{1001, 0101, 0011\}$, although there are other possibly more natural choices.

Example 5.2. Let

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix},$$

and let C be the binary 6-bit linear code spanned by the rows of A. That is, $C = \mathrm{row}(A)$. Since A is in row reduced form, its rows are independent, hence form a set of basic codewords for C. Thus C is a three dimensional subspace of $V(6,2)$, so $|C| = 8$. The 7 non zero codewords are

$$(100111),\ (010101),\ (001011),\ (110010),\ (101100),\ (011110),\ (1111001).$$

Note that all possible combinations of 0's and 1's occur in the first three positions. These three letters tell you which linear combination of the basic codewords is involved. The last three letters are again check digits.

5.2.2 The International Standard Book Number

The International Standard Book Number (ISBN) is a reference number that is issued to books published by the mainstream publishing companies. Its purpose is to assist bookstores in making orders and to help librarians in cataloguing. The system has been in place since 1969. Each ISBN is a 10 digit string $a_1 \cdots a_9\ a_{10}$. The digits $a_1, \ldots, a_9$ are allowed to take any value between 0 and 9, but the last digit a_{10} can also take the value X, which is the Roman numeral denoting 10.

For example, the book *Fermat's Enigma* by Simon Singh, published in 1997 by Penguin Books, has ISBN 0-14-026869-3. The first digit 0 indicates that the book is in English, the digits between the first and second hyphens give the number assigned to the publisher, and the next set of digits indicates the title. The last digit is the check digit, which we will explain below. Major publishing companies like Penguin have small numbers (Penguin's is 14), while small publishers are given a larger number. Whitecap Books in Vancouver and Toronto has the 6 digit number 921061. Thus Penguin can publish 999,999 titles (in English), but Whitecap is restricted to 99.

ISBN's are based on a linear 11-ary [10,9] code, that is, a 9-dimensional linear subspace C of $V(10, 11)$. The code C is defined to be the solution space of the homogeneous linear equation in $a_1, \ldots, a_{10}$ given by

$$a_1 + 2a_2 + 3a_3 + \cdots + 9a_9 + 10a_{10} = 0.$$

Clearly, C can also be described as the null space of the rank one matrix $(1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10)$ over $\mathbb{F}_{11}$. Since $10+1 = 0$ in $\mathbb{F}_{11}$, the defining equation can also be expressed as

$$a_{10} = \sum_{i=1}^{9} i a_i.$$

The ISBN's are the codewords

$$a_1\ a_2\ a_3\ a_4 \ldots a_9\ a_{10}$$

described above (with hyphens inserted in appropriate places). Of course, not all 11^9 possible codewords can be used because of the restriction $a_i \neq 10$ except for a_{10}.

Example 5.3. For example, 0-15-551005-3 is an ISBN since $0 + 2 + 15 + 20 + 25 + 6 + 0 + 0 + 453 \equiv 3 \bmod(11)$, as is 0-14-026869-3 from the above example.

Example 5.4. Suppose that an ISBN is entered as 0-19-432323-1. With a minimum amount of technology, the machine in which the numbers are being entered will warn the librarian that 0-19-432323-1 is not an ISBN: that is, $(0, 1, 9, 4, 3, 2, 3, 2, 3)$ doesn't satisfy $\sum_{i=1}^{10} ia_i = 0$ in $\mathbb{F}_{11}$. Thus an error has been detected. But the type of error isn't. For example, there may be a single incorrect digit, or two digits might have been transposed. In fact, these two possibilities are the most common types of error. The next result says something about them.

Proposition 5.3. *A vector $\mathbf{a} = (a_1, \ldots, a_{10}) \in V(10, 11)$ that differs from an element of C in exactly one place cannot belong to C; in particular it cannot be an ISBN. Similarly, an element of $V(10, 11)$ obtained by transposing two unequal letters of an ISBN cannot be an ISBN.*

Proof. We will prove the first assertion but leave the second as an exercise. Suppose $\mathbf{c} = (c_1, \ldots, c_{10})$ is a codeword which differs from $\mathbf{a} \in V(10, 11)$ in one exactly component, say $c_i = a_i$ if $i \neq j$, but $c_j \neq a_j$. Then

$$\mathbf{v} := \mathbf{a} - \mathbf{c} = (0, \ldots, 0, a_j - c_j, 0, \ldots, 0).$$

If $\mathbf{a} \in C$, then $\mathbf{v} \in C$ too, hence $j(a_j - c_j) = 0$ in $\mathbb{F}_{11}$. But since neither j nor $a_j - c_j$ is zero in $\mathbb{Z}_{11}$, this contradicts the fact that $\mathbb{F}_{11}$ is a field. Hence $\mathbf{v} \notin C$, so $\mathbf{a} \notin C$ also. This proves the first assertion. □

Suppose you know all but the kth digit of an ISBN. Can you find the missing digit? Try this with an example, say 0-13-832x44-3. This is a sure way to astound your friends and relatives and maybe win a few bets. But don't bet with a librarian.

Exercises

Exercise 5.1. Determine all x such that 0-13-832x4-4 is an ISBN.

Exercise 5.2. Determine all x and y such that both 1-2-3832xy4-4 and 3-33-x2y377-6 are ISBNs.

Exercise 5.3. Prove the second assertion of Proposition 5.3.

5.3 Error detecting and correcting codes

5.3.1 Hamming Distance

In $\mathbb{R}^n$, the distance between two vectors is the square root of the sum of the squares of the differences of their components. This could never be used

to measure the distance between two elements of $V(n,p)$ since a sum of squares in $\mathbb{F}_p$ may well be 0. It turns out however that there is another way of measuring distances and lengths which works extremely well in the $V(n,p)$ setting.

Definition 5.4. Suppose $\mathbf{v} = (v_1, \ldots, v_n) \in V(n,p)$. Define the *weight* $\omega(\mathbf{v})$ of $\mathbf{v}$ to be the number of i such that $v_i \neq 0$. That is,

$$\omega(v_1, \ldots, v_n) = |\{i \mid v_i \neq 0\}|.$$

The *Hamming distance* (or simply the distance) $d(\mathbf{u}, \mathbf{v})$ between $\mathbf{u}$ and $\mathbf{v}$ in $V(n,p)$ is defined by

$$d(\mathbf{u}, \mathbf{v}) = \omega(\mathbf{u} - \mathbf{v}).$$

For example, $\omega(1010111) = 5$. Note that the only vector of weight zero is the zero vector. Therefore $\mathbf{u} = \mathbf{v}$ exactly when $\omega(\mathbf{u} - \mathbf{v}) = 0$. What makes the Hamming distance d so important is that it satisfies the three properties (i), (ii) and (iii) below which define what is called a *metric on* $V(n,p)$.

Proposition 5.4. *Suppose* $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V(n,p)$. *Then:*

(i) $d(\mathbf{u}, \mathbf{v}) \geq 0$, *and* $d(\mathbf{u}, \mathbf{v}) = 0$ *if and only if* $\mathbf{u} \neq \mathbf{v}$;

(ii) $d(\mathbf{u}, \mathbf{v}) = d(\mathbf{v}, \mathbf{u})$; *and*

(iii) $d(\mathbf{u}, \mathbf{w}) \leq d(\mathbf{u}, \mathbf{v}) + d(\mathbf{v}, \mathbf{w})$.

These properties clearly hold for the usual metric $d(\mathbf{u}, \mathbf{v}) = |\mathbf{u} - \mathbf{v}|$ on $\mathbb{R}^n$. Property (iii) is known as the *triangle inequality*, so named because in $\mathbb{R}^n$ it says that the length of any side of a triangle can't exceed the sum of the lengths of the other two sides. The first two properties of the Hamming distance are easy to see, but the triangle inequality requires proof.

Proof. First consider the case where $\mathbf{u}$ and $\mathbf{v}$ differ in every component. Thus $d(\mathbf{u}, \mathbf{v}) = n$. Let $\mathbf{w}$ be any vector in $V(n,p)$, and suppose $d(\mathbf{u}, \mathbf{w}) = k$. Then $\mathbf{u}$ and $\mathbf{w}$ agree in $n - k$ components, which tells us that $\mathbf{v}$ and $\mathbf{w}$ cannot agree in those $n - k$ components, so $d(\mathbf{v}, \mathbf{w}) \geq n - k$. Thus

$$d(\mathbf{u}, \mathbf{v}) = n = k + (n - k) \leq d(\mathbf{u}, \mathbf{w}) + d(\mathbf{v}, \mathbf{w}).$$

In the general case, let $\mathbf{u}, \mathbf{v}, \mathbf{w}$ be given, and let $\mathbf{u}', \mathbf{v}'$ and $\mathbf{w}'$ denote the vectors obtained by dropping the components where $\mathbf{u}$ and $\mathbf{v}$ agree. Thus we are in the previous case. Now

$$d(\mathbf{u}, \mathbf{v}) = d(\mathbf{u}', \mathbf{v}') \leq d(\mathbf{u}', \mathbf{w}') + d(\mathbf{u}, \mathbf{w}').$$

But $d(\mathbf{u}', \mathbf{w}') \leq d(\mathbf{u}, \mathbf{w})$ and $d(\mathbf{v}', \mathbf{w}') \leq d(\mathbf{v}, \mathbf{w})$. Therefore,

$$d(\mathbf{u}, \mathbf{v}) \leq d(\mathbf{u}, \mathbf{w}) + d(\mathbf{v}, \mathbf{w}),$$

and the triangle inequality is established. □

One of the most desirable features for a (not necessarily linear) code C is that the minimum distance between two any codewords as large as possible. Let $d(C)$ denote this minimum distance. For a code which isn't linear, $d(C)$ has to be computed in the old fashioned way, that is the distance between every pair of codewords has to be taken into account. In general, if there are m codewords, then this means doing

$$\binom{m}{2} = \frac{m(m-1)}{2}$$

calculations (check this). But if C is linear, finding $d(C)$ takes much less.

Proposition 5.5. *If $C \subset V(n, 2)$ is linear, then $d(C)$ is the minimum of the weights of all the non zero codewords.*

We will leave the proof as an exercise.

5.3.2 The Main Result

An code $C \subset V(n, p)$ of length n such that $|C| = M$ and $d(C) = d$ is often called an (n, M, d)-code. If C is also linear, $M = p^k$ for some positive integer k. In that case, we say that C is a *p-ary $[n, k, d]$-code*. In general, one wants to maximize $d(C)$. The reason for this is given in the next result.

Proposition 5.6. *A (not necessarily linear) (n, M, d)-code C can detect up to $d - 1$ errors, i.e. if $d(\mathbf{v}, \mathbf{c}) \leq d - 1$ for some $\mathbf{c} \in C$, then $\mathbf{v} \notin C$. Moreoever, C corrects up to $e = (d - 1)/2$ errors. That is, if $d(\mathbf{v}, \mathbf{c}) \leq e$, for some codeword $\mathbf{c}$, then this $\mathbf{c}$ is the unique codeword with this property, and thus $\mathbf{c}$ corrects the errors in the non-codeword $\mathbf{v}$.*

The error-correcting assertion can be succinctly phrased by saying that any $\mathbf{v}$ within Hamming distance $e = (d - 1)/2$ of C is within e of a unique codeword. So if you know all but e digits of a codeword, you know them all.

Example 5.5. Suppose C is a 6-bit code with $d = 3$. Then $e = 1$. If $\mathbf{c} = 100110$ is a codeword, then $\mathbf{v} = 000110$ can't be one, but 100110 is the unique codeword within Hamming distance 1 of the non-codeword 000110.

We will leave the first assertion of Proposition 5.6 as an exercise and prove the harder second assertion. Assume $d(\mathbf{v}, \mathbf{c}) \leq (d-1)/2$, and suppose there exists an $\mathbf{c}' \in C$ such that $d(\mathbf{v}, \mathbf{c}') \leq d(\mathbf{v}, \mathbf{c})$. Thus,

$$d(\mathbf{v}, \mathbf{c}') \leq d(\mathbf{c}, \mathbf{v}) \leq (d-1)/2.$$

The idea is to use the triangle identity to estimate $d(\mathbf{c}, \mathbf{c}')$, which we know is at least $d(C) = d$ if $\mathbf{c} \neq \mathbf{c}'$. But by the triangle inequality,

$$d(\mathbf{c}, \mathbf{c}') \leq d(\mathbf{c}, \mathbf{v}) + d(\mathbf{v}, \mathbf{c}') \leq (d-1)/2 + (d-1)/2 = d-1,$$

so indeed we have, we have $\mathbf{c} = \mathbf{c}'$. □

For the binary [4,3]-code given by $x_1 + x_2 + x_3 + x_3 + x_4 = 0$, one sees easily that $d(C) = 2$. Thus C detects a single error, but can't correct an error because $(d-1)/2 = 1/2 < 1$. However, if some additional information is known, such as the component where the error occurs, it can be corrected using the linear equation defining the code.

5.3.3 Perfect Codes

We can also interpret Proposition 5.6 geometrically. If $r > 0$, define the *ball of radius r centred at $\mathbf{v} \in V(n,p)$* to be

$$B_e(\mathbf{v}) = \{\mathbf{w} \in V(n,p) \mid d(\mathbf{w}, \mathbf{v}) \leq e\}. \tag{5.1}$$

Extending Proposition 5.6, we show

Proposition 5.7. *Let $C \subset V(n,p)$ satisfy $d(C) = d$, and let $e = (d-1)/2$. For any $\mathbf{c} \in C$,*

$$B_e(\mathbf{c}) \cap C = \{\mathbf{c}\}.$$

Hence, an element $\mathbf{v} \in V(n,p)$ which lies in one of the balls $B_e(\mathbf{c})$ lies in exactly one of them.

Proof. This follows immediately from Proposition 5.6. □

Of course, the Proposition doesn't say much unless $d(C) > 2$. Thus the union of the balls $B_e(\mathbf{c})$ as $\mathbf{c}$ varies over C is the set of elements of $V(n,p)$ which are within e of a unique codeword. The nicest situation is that these balls cover $V(n,p)$, that is

$$V(n,p) = \bigcup_{\mathbf{c} \in C} B_e(\mathbf{c}). \tag{5.2}$$

Definition 5.5. A code $C \subset V(n,p)$ with $d(C) = d > 2$ is said to be *perfect* if (5.2) holds. That is, C is perfect if and only if every element of $V(n,p)$ is within e of a (unique) codeword.

We will consider perfect codes in more detail in §5.15. It turns out that perfect codes are not so abundant. There are infinitely many perfect binary linear codes C with $d(C) = 3$, hence $e = 1$. These single error correcting codes are known as *Hamming codes*. A result of Pless says that there are only two perfect linear codes with $e > 1$. One is a binary $[23, 12]$-code with $d = 7$ and the other is a ternary $[11, 6]$-code with $d = 5$.

5.3.4 A Basic Problem

One of the basic problems in coding theory is to design codes $C \subset V(n,p)$ such that both $|C|$ and $d(C)$ are large. More precisely, the problem is to maximize $|C|$ among all codes C for which $d(C) \geq m$ for some given integer m. The maximum will then depend on n and m. If we also impose the condition that C is linear, then we are actually seeking to maximize $\dim(C)$, since $|C| = p^{\dim(C)}$. An example which has this property is the binary $(8, 16, 4)$ code C_8 defined in the next example.

Example 5.6. Consider the following matrix

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

The row space C_8 of A is called the *extended Hamming code*. Notice that every row of A has weight 4, so the minimum distance of C_8 is at most 4. In fact, it can be shown that $d(C_8) = 4$. Hence C_8 is an $(8, 16, 4)$-linear code.

Proposition 5.8. *The code C_8 maximizes $|C|$ among all 8-bit binary linear codes with $d(C) \geq 4$.*

Proof. Since $\dim C_8 = 4$, we have to show that there are no 8-bit binary linear codes C with $d(C) \geq 4$ and $|C| > 16$. Suppose C is in fact one such code. Then by taking a spanning set for C as the rows of a $k \times 8$ matrix A, we can use row operations to put A into reduced row echelon form A_{red} without changing C. For simplicity, suppose that A_{red} has the form $(I_r|M)$. It follows that $|C| = 2^r$, so since $|C| > 16$, we see that $r \geq 5$. Hence M has at most three columns. Now the only way $d(C) \geq 4$ is if all entries of M

are 1. But then subtracting the second row of A_{red} from the first gives an element of C of weight 2, which contradicts $d(C) \geq 4$. Thus $r \leq 4$. □

In fact, by a similar argument, we can show the *singleton bound* for $d(C)$.

Proposition 5.9. *If C is a linear $[n, k]$-code, then*

$$d(C) \leq n - k + 1.$$

Put another way, a linear code C of length n satisfies

$$\dim C + d(C) \leq n + 1.$$

We leave the proof as an exercise. In the next section, we will consider a class of non-linear binary where both $|C|$ and $d(C)$ are large. Let us a make a final definition.

Definition 5.6. A linear $[n, k]$-code C with $d(C) = n - k + 1$ is said to be *maximal distance separating.*

5.3.5 Linear Codes Defined by Generating Matrices

The purpose of this subsection is to consider linear codes C given as the row space of a so called *generating matrix*. We already considered some examples of generating matrices in the last subsection.

Definition 5.7. A *generating matrix* for a linear $[n, k]$-code C is a $k \times n$ matrix over $\mathbb{F}_p$ of the form $M = (I_k \mid A)$ such that $C = \text{row}(M)$.

Example 5.7. Let C be the binary [4,2]-code with generating matrix

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

Taking all the linear combinations of the rows, we find that

$$C = \{0000, 1011, 0101, 1110\}.$$

A check to make sure that we have found all of C is to note that since $\dim C = 2$ and $p = 2$, C has $4 = 2^2$ elements.

Example 5.8. Let

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

One checks easily that besides the rows of M, the elements of C are

$$(000000),\ (110010),\ (101100),\ (011110),\ (111001).$$

Clearly $d(C) = 3$, so C is one error-correcting .

The reader should also recall the code C_8 considered in Example 5.6. Every element of $C = \text{row}(M)$ can be expressed as a matrix product of the form $(x_1\ \dots\ x_k)M$. (To see this, transpose the fact that the column space of M^T consists of all vectors of the form $M^T(y_1\ \dots\ y_n)^T$.) Now, to any $\mathbf{x} = (x_1 \dots x_k) \in \mathbb{F}^k$, there is a unique codeword $\mathbf{c}(\mathbf{x}) = (x_1\ \dots\ x_k)M \in C$. For a generating matrix M as above,

$$\mathbf{c}(\mathbf{x}) = x_1 \dots x_k \sum_{i=1}^{k} a_{i1}x_i \cdots \sum_{i=1}^{k} a_{i(n-k)}x_i.$$

Since $x_1, \dots, x_k$ are completely arbitrary, the first k entries $x_1 \dots x_k$ are called the *message digits* and the last $n-k$ digits are called the *check digits*.

Exercises

Exercise 5.4. Prove the second assertion of Proposition 5.3.

Exercise 5.5. Prove the first two parts of Proposition 5.5.

Exercise 5.6. Consider the binary code $C \subset V(6,2)$ which consists of 000000 and the following nonzero codewords:

$$(100111), (010101), (001011), (110010), (101100), (011110), (111001).$$

(i) Determine whether or not C is linear.

(ii) Compute $d(C)$.

(iii) How many elements of C are nearest to (011111)?

(iv) Determine whether or not 111111 is a codeword. If not, is there a codeword nearest 111111?

Exercise 5.7. Prove the first part of Proposition 5.6.

Exercise 5.8. Compute $d(C)$ for the code C of Example 5.2.

Exercise 5.9. Consider the binary code C_7 defined as the row space of the matrix

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

in $V(7,2)$.

(i) Compute $d(C)$ and e.

(ii) Find the unique element of C that is nearest to 1010010. Do the same for 1110001.

Exercise 5.10. Let r be a positive integer and consider the ball $B_r(\mathbf{x}) \subset V(n,2)$ about $\mathbf{x} \in V(n,2)$. Show that

$$|B_r(\mathbf{x})| = \sum_{i=0}^{r} \binom{n}{i}.$$

Exercise 5.11. Generalize Exercise 5.10 from $V(n,2)$ to $V(n,p)$.

Exercise 5.12. * Show that if C is a linear binary $[n,k]$-code and C is e-error-correcting , then

$$\sum_{i=0}^{e} \binom{n}{i} \leq 2^{(n-k)}.$$

In particular, if C is 1-error-correcting , then $|C| \leq 2^n/(1+n)$.

Exercise 5.13. Show that if P is a permutation matrix, then P defines a transformation $T : V(n,p) \to V(n,p)$ which preserves the Hamming distance.

Exercise 5.14. Show that if C is a linear code, then

$$d(C) = \min\{\omega(\mathbf{x}) \mid \mathbf{x} \in C, \mathbf{x} \neq \mathbf{0}\}.$$

That is, $d(C)$ is the minimum weight among the non zero vectors in C. Use the result to find $d(C)$ for the code C used to define ISBN's? Is this code error-correcting ?

Exercise 5.15. Prove Proposition 5.9. (Note,

Exercise 5.16. Taking $\mathbb{F} = \mathbb{F}_{11}$, compute the generating matrix for the ISBN code.

5.4 Hadamard matrices

We will next consider an interesting class of binary codes based on Hadamard matrices, which are named after the French mathematician J. Hadamard. As mentioned above, these so called *Hadamard codes* have the property that both $|C|$ and $d(C)$ have a large values (as opposed to what we saw in Proposition 5.9 for linear codes). Hadamard matrices are themselves of interest, since their properties are not that well understood. As Hadamard codes are nonlinear, we consider this topic to be a sightseeing trip.

5.4.1 Hadamard Matrices

A *Hadamard matrix* is an $n \times n$ matrix H such that $h_{ij} = \pm 1$ for all $1 \leq i, j \leq n$ and

$$HH^T = nI_n.$$

Proposition 5.10. *If H is an $n \times n$ Hadamard matrix, then:*

(i) $H^T H = nI_n$,

(ii) any two distinct rows or any two distinct columns are orthogonal;

(iii) n is either 1, 2 or a multiple of 4; and

(iv) if $n > 1$, then any two rows of H agree in exactly $n/2$ places.

The only assertion that isn't obvious is (iii). We will omit the proof. It's still an open problem as to whether there is a $4k \times 4k$ Hadamard matrix for every $k > 0$. This is known for $k \leq 106$, but it doesn't seem to be known whether there is a 428×428 Hadamard matrix.

Example 5.9. Examples of $n \times n$ Hadamard matrices for $n = 2, 4, 8$ are

$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad H_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix},$$

and

$$H_8 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \end{pmatrix}.$$

After this point, it is no longer instructive to write them down. One can produce other Hadamard matrices from these by the transformation $H \mapsto PHQ$, where P and Q are permutation matrices.

5.4.2 Hadamard Codes

We will now define a Hadamard code. Let H be any $n \times n$ Hadamard matrix. Consider the $n \times 2n$ matrix $(H| - H)$, and let $\mathcal{H}$ be the binary matrix obtained by replacing all -1's by 0's.

Definition 5.8. The *Hadamard code* C associated to H is by definition the set of columns of $\mathcal{H}$. It is a binary n-bit code with $2n$-codewords.

Proposition 5.11. *Let C be an n-bit Hadamard code. Then $d(C) = n/2$. Thus C is a binary $(n, 2n, n/2)$-code.*

Proof. Recall that n is a multiple of 4, so $n/2$ is an even integer. The fact that the ith and jth columns of H are orthogonal if $i \neq j$ implies they must differ in exactly $n/2$ components since all the components are ± 1. But the ith and jth columns of H and $-H$ are also orthogonal if $i \neq j$, so they differ in $n/2$ places too. Moreover, the ith columns of H and $-H$ differ in n places. This proves $d(C) = n/2$, as asserted. □

For example, the Hadamard matrix H_2 gives the $(2, 4, 1)$-code

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

The code that was used in the transmission of data from the Mariner space probes to Venus in the 1970's was a binary $(32, 64, 16)$ Hadamard code. Since $(16-1)/2 = 7.5$, this code corrects 7 errors. An interesting footnote is that the Mariner space probes, now billions and billions of miles from earth, are still transmitting data.

5.5 The Standard Decoding Table, Cosets and Syndromes

5.5.1 The Nearest Neighbour Decoding Scheme

Suppose that $C \subset V(n,p)$ is a p-ary linear n-bit code used in transmitting data from a satellite. Assume a codeword $\mathbf{c} = c_1 \dots c_n$ has been transmitted and received as $\mathbf{d} = d_1 \dots d_n$. Due to atmospheric interference and a number of other possible sources of noise, the received word $\mathbf{d}$ is not a codeword. The people manning the communication center therefore have to consider the *error* $\mathbf{E} := \mathbf{d} - \mathbf{c}$, which of course is unknown to them. The object is to make an intelligent guess as to what $\mathbf{c}$ is.

One popular scheme is the *nearest neighbour* decoding scheme, which uses a *standard decoding table* for C (SDT for short). The idea is to organize the elements of $V(n,p)$, into the cosets of C, which were defined in Definition 4.6. For the convenience of readers who skipped the notion of a quotient space, we will recall the basic properties of cosets below. The cosets of C are disjoint subsets of $V(n,p)$ whose union is all of $V(n,p)$. Any two elements in the same coset are said to have the same error. Put precisely, the cosets of C were defined to be the equivalence classes of the equivalence relation given on $V(n,p)$ given by $\mathbf{x}E\mathbf{y}$ if and only if $\mathbf{x} - \mathbf{y} \in C$. (Thus cosets are analogous to the set of all parallel planes in $\mathbb{R}^3$.) This is illustrated in the following example.

Example 5.10. Let $C \subset V(4,2)$ be the linear code of Example 5.7. We will now construct an SDT for C. The SDT is a rectangular array listing all 2^4 elements of $V(4,2)$ as follows. The first row consists of the elements of C, putting $\mathbf{0} = 0000$ in the first column. To construct the second row, choose an element $\mathbf{E}_1$ not in C, and put $\mathbf{E}_1$ directly below 0000. Now add $\mathbf{E}_1$ to each element $\mathbf{c}$ in the first row to the right of 0000 and write the result $\mathbf{E}_1 + \mathbf{c}$ directly below $\mathbf{c}$. Thus the first row is the coset $\mathbf{0} + C$ of $\mathbf{0}$, and the second row is the coset $\mathbf{E}_1 + C$ of $\mathbf{E}_1$. Next, select a $\mathbf{E}_2 \in V(4,2)$ which isn't in either of the first two rows, assuming one exists, and repeat the previous step with $\mathbf{E}_2$. Continuing this construction will eventually exhaust $V(4,2)$, and the final result is the standard decoding table . This construction, however, may lead to many different standard decoding tables since there aren't necessarily any canonical choices for the error vectors that appear in

the first column. Below is an example of a standard decoding table for C.

$$\begin{array}{cccc} 0000 & 1011 & 0101 & 1110 \\ 1000 & 0011 & 1101 & 0110 \\ 0100 & 1111 & 0001 & 1010 \\ 0010 & 1001 & 0111 & 1100 \end{array} .$$

Every row of a standard decoding table for a subspace $C \subset V(n,p)$ has the form $\mathbf{E} + C$ for some $\mathbf{E} \in V(n,p)$. The first row is $C = \mathbf{0} + C$, and the potential errors $\mathbf{E}_i$ we've selected vary through the first column. One obvious comment is that it makes sense to choose errors of minimal weight. If a codeword $\mathbf{c}$ has been transmitted but a non-codeword such as 0111 is received, then scan the standard decoding table until 0111 is located. In the example, 0111 occurs in the last row directly below the codeword 0101. The nearest neighbour decoding scheme assumes that the error is the leading element 0010 of the last row, so the correct codeword is $0101 = 0111 - 0010$.

Notice that it can happen that a row of a standard decoding table contains more than one element of minimal weight. This happens in the third row of the above table, where there are two elements of weight one. There is no reason to prefer decoding 1111 as 1011 rather than 1110. The non zero elements of least nonzero weight in $V(n,2)$ are standard basis vectors. If $\dim C = k$, then at most k of the standard basis vectors can lie in C. These vectors are therefore natural candidates for the leading column. In fact, it seems desirable to seek codes C so that there is a standard decoding table such that in each row, there is a unique vector of minimal length. We will see presently that this objective is achieved by perfect linear codes.

5.5.2 Cosets

From an inspection of the above standard decoding table, three properties are apparent:

(a) different rows don't share any common elements;

(b) any two rows have the same number of elements; and

(c) every element of $V(4,2)$ is in some row.

These properties follow from the fact that the rows of a standard decoding table are the cosets s of C.

Definition 5.9. Let V be a vector space over $\mathbb{F}$, and suppose A and B are subsets of V. We define $A + B$ to be the subset consisting of all vectors of the form $a + b$, where $a \in A$ and $b \in B$. If C is a subspace of V, then a subset of V of the form $\{\mathbf{v}\} + C$ is called a *coset of* C.

To simplify the notation, we will denote $\{\mathbf{v}\}+C$ by $\mathbf{v}+C$. For example, each row of a standard decoding table is a coset of the linear code C, since it has the form $\mathbf{E}+C$. The properties (a), (b) and (c) stated above all follow from

Proposition 5.12. *Let V be a vector space over $\mathbb{F}_p$ of dimension n, and let C be a linear subspace of V of dimension k. Every element of V lies in a coset of C, and two cosets are either disjoint or equal. In fact, $\mathbf{v}+C = \mathbf{w}+C$ if and only if $\mathbf{w} - \mathbf{v} \in C$. Finally, there are $p^{(n-k)}$ cosets of C, every coset contains p^k elements.*

Proof. Certainly $\mathbf{v} \in \mathbf{v} + C$, so the first claim is true. If $\mathbf{w} + C$ and $\mathbf{v} + C$ contain an element $\mathbf{y}$, then $\mathbf{y} = \mathbf{w} + \mathbf{c} = \mathbf{v} + \mathbf{d}$ for some $\mathbf{c}, \mathbf{d} \in C$. Thus $\mathbf{w} = \mathbf{v} + \mathbf{c} - \mathbf{d}$. Since $\mathbf{c} - \mathbf{d} \in C$, it follows that $\mathbf{w} + C = \mathbf{v} + (\mathbf{c} - \mathbf{d}) + C$. But since C is a subspace, $(\mathbf{c} - \mathbf{d}) + C = C$, so $\mathbf{w} + C = \mathbf{v} + C$. This proves the second claim. If $\mathbf{v} + C = \mathbf{w} + C$, then $\mathbf{w} - \mathbf{v} \in C$, and conversely. To prove the last assertion, recall that $|C| = p^k$. Hence $|\mathbf{v} + C| = p^k$ too. For $\mathbf{v} \to \mathbf{v} + \mathbf{c}$ is a bijection from C to $\mathbf{v} + C$. It follows that there are $p^{(n-k)}$ cosets, which completes the proof. □

In coding theory, the error elements $\mathbf{E}$ in the first column of a particular standard decoding table are sometimes called *coset leaders*, although in other contexts, they are known as coset representatives..

5.5.3 Syndromes

One can modify the construction of an standard decoding table so that it isn't necessary to scan the whole table to find a given entry. This is important since scanning is an inefficient process in terms of computer time. The amount of scanning can be greatly reduced by using syndromes. The simplification come about by introducing the notion of a parity check matrix.

Definition 5.10. Let $C \subset V(n, p)$ be a linear code given by a generating matrix $M = (I_k \mid A)$. A *parity check matrix* for C is an $(n-k) \times n$ matrix H such that C is the null space of H after identifying row and column vectors.

One of the advantages of using a generating matrix in the form $M = (I_k \mid A)$ is that the parity check matrix H is simple to write down.

Proposition 5.13. *Suppose $C \subset V(n,p)$ is the code obtained as the row space of a generating matrix $M = (I_k \mid A)$. Then $\mathbf{c} \in C$ if and only if $(-A^T \mid I_{n-k})\mathbf{c}^T = \mathbf{0}$. Thus, $H = (-A^T \mid I_{n-k})$ is a parity check matrix for C. Furthermore, two vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ in $V(n,p)$ are in the same coset of C if and only if $H\mathbf{d}_1^T = H\mathbf{d}_2^T$.*

We will leave the proof that H is a parity check matrix as an exercise. When C is a binary code, the parity check matrix $H = (A^T \mid I_{n-k})$, since $-A^T = A^T$.

We now give the proof of the second assertion. Note that $\mathbf{d}_1$ and $\mathbf{d}_2$ are in the same coset of C if and only if $\mathbf{d}_1 - \mathbf{d}_2 \in C$ if and only if $H(\mathbf{d}_1 - \mathbf{d}_2)^T = \mathbf{0}$ if and only if $H\mathbf{d}_1^T = H\mathbf{d}_2^T$. □

Definition 5.11. We call $\mathbf{d}H^T = (H\mathbf{d}^T)^T$ the *syndrome* of $\mathbf{d} \in V(n,p)$ with respect to C.

To incorporate syndromes in a standard decoding table , we insert an extra column consisting of the syndromes of the cosets. Thus each row consists of a coset and the syndrome of that coset. The different syndromes identify the different cosets, so instead of having to scan the whole decoding table to find $\mathbf{d}$, it suffices to first scan the column of syndromes to find the syndrome of $\mathbf{d}$ and then scan that row.

Example 5.11. Let M be the generating matrix of Example 5.7. Recall

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

Thus

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

The syndromes are found by taking the matrix product

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

Thus the standard decoding table with syndromes is

0000	1011	0101	1110	00
1000	0011	1101	0110	11
0100	1111	0001	1010	01
0010	10001	0111	1100	10

Exercises

Exercise 5.17. Construct the standard decoding table with syndromes for the binary code C with generating matrix

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

(Note, this is rather complicated since C has 8 elements. Thus the standard decoding table with syndromes is a 8×9 table (counting the syndromes as one column) since $V(6,2)$ has 64 elements. Perhaps you can write a program.)

Exercise 5.18. Let C be the code of the previous problem.

(a) How many errors does C detect?

(b) How many errors does it correct?

(c) Use the standard decoding table with syndromes you constructed in Exercise 5.18 to decode 101111 and 010011.

Exercise 5.19. Show that, indeed, $C = \{\mathbf{c} \in V(n,2) \mid \mathbf{c}H^T = \mathbf{0}\}$. (Suggestion: begin by showing that $MH^T = \mathbf{O}$. Hence every row of M lies in the left null space of H. Now compute the dimension of the left null space of H, using the fact that H and H^T have the same rank.)

Exercise 5.20. Construct the standard decoding table for the binary code with generating matrix

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

Exercise 5.21. Let C be a binary linear n-bit code with $n \geq 3$ with parity check matrix H. Show that if no two columns of H are dependent (i.e. equal), then $d(C) \geq 3$.

Exercise 5.22. Generalize Exercise 5.21 by showing that if $C \subset V(n,p)$ is a linear code with a parity check matrix H having the property that no m columns of H are linearly dependent, then $d(C) \geq m+1$.

Exercise 5.23. Show that any coset of the ISBN contains a unique standard basis vector. In particular, any 10 digit number differs form an ISBN in one digit. Can you determine in general when it is possible, for a given C, to have a standard decoding table where the coset leaders are the standard basis vectors?

Exercise 5.24. Find an upper bound on the number of operations required to scan the standard decoding table (with syndromes) associated to an *p-ary* $[n, k]$ code to find any $\mathbf{d} \in V(n, 2)$. Compare this result to the number of operations needed to find $\mathbf{d}$ before adding the syndromes.

Exercise 5.25. Complete the proof of Proposition 5.13 by showing that H is a parity check matrix.

5.6 Perfect linear codes

Recall from Definition 5.5 that a code $C \subset V(n,p)$ with $d(C) = d > 2$ is perfect if and only if

$$V(n,p) = \bigcup_{\mathbf{c}\in C} B_e(\mathbf{c}). \tag{5.3}$$

The purpose of this section is to find a condition for the perfection of a linear code. First, let us note a property of the Hamming distance. Namely, d is what is called a *translation invariant metric*. That is, for all $\mathbf{w}, \mathbf{x}, \mathbf{y} \in V(n,p)$, we have

$$d(\mathbf{w}+\mathbf{y}, \mathbf{x}+\mathbf{y}) = d(\mathbf{w}, \mathbf{x}).$$

Indeed,

$$d(\mathbf{w}+\mathbf{y}, \mathbf{x}+\mathbf{y}) = \omega(\mathbf{w}+\mathbf{y}-(\mathbf{x}+\mathbf{y})) = \omega(\mathbf{w}-\mathbf{x}) = d(\mathbf{w}, \mathbf{x}).$$

We will prove the main result on perfect linear codes below. Before doing this, however, we want to prove a fact about the ball $B_e(\mathbf{0})$ which will be used in the next section. I claim that a coset $\mathbf{x}+C$ cannot meet a $B_e(\mathbf{0})$ in more than one point. For if $\mathbf{x}+\mathbf{c}$ and $\mathbf{x}+\mathbf{c}'$ both lie in $B_e(\mathbf{0})$, then by translation invariance and the triangle inequality,

$$d(\mathbf{c}, \mathbf{c}') = d(\mathbf{x}+\mathbf{c}, \mathbf{x}+\mathbf{c}') \leq d(\mathbf{x}+\mathbf{c}, \mathbf{0}) + d(\mathbf{0}, \mathbf{x}+\mathbf{c}') \leq 2e < d,$$

and this implies $\mathbf{c} = \mathbf{c}'$. Hence we get

Proposition 5.14. *For a linear code $C \subset V(n,p)$ with minimal distance d, $|B_e(\mathbf{0})| \leq p^{n-k}$, where $k = \dim C$.*

Proof. By the above claim, we simply have to recall from Proposition 5.12 that the number of cosets of C is p^{n-k}. □

Here is the main result.

Theorem 5.15. *A linear code $C \subset V(n,p)$ with $d > 2$ is perfect if and only if every coset $\mathbf{x}+C$ of C contains a unique element of $B_e(\mathbf{0})$.*

Proof. Suppose C is perfect, and consider a coset $\mathbf{x}+C$. By (5.3), $\mathbf{x}+C$ meets some $B_e(\mathbf{c})$, say $\mathbf{x}+\mathbf{c}' \in B_e(\mathbf{c})$. Thus $d(\mathbf{x}+\mathbf{c}', \mathbf{c}) = d(\mathbf{x}+(\mathbf{c}'-\mathbf{c}), \mathbf{0}) \leq e$, by translation invariance. Hence $\mathbf{x}+(\mathbf{c}'-\mathbf{c}) \in B_e(\mathbf{0})$, so $\mathbf{x}+C$ meets $B_e(\mathbf{0})$. By the remark preceding the Theorem, it follows that every coset $\mathbf{x}+C$ of C contains a unique element of $B_e(\mathbf{0})$.

To prove the converse, suppose $\mathbf{y} \in V(n,p)$. Let $\mathbf{x} = \mathbf{y}+\mathbf{c}$ be the element of $y+C$ contained in $B_e(\mathbf{0})$. Then $d(\mathbf{x}, \mathbf{0}) = d(\mathbf{x}-\mathbf{c}, -\mathbf{c}) \leq e$, so $\mathbf{y} = \mathbf{x}-\mathbf{c} \in B_e(-\mathbf{c})$. Hence C is perfect. □

It follows immediately from the Theorem that if C is a perfect linear code, then there exists a standard decoding table for C, which is unique up to how we enumerate C, such that each coset leader lies in $B_e(\mathbf{0})$.

Example 5.12. The binary code $C_3 = \{000, 111\}$ is perfect. Note $d = 3$ so $e = 1$. Thus the perfection is clear since any element of $V(3, 2)$ is one unit away from a codeword (namley 000 if it has two 0's and 111 if it has two 1's. The generating matrix of C_3 is $M = (I_1|1\,1)$. Thus the parity check matrix is

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

The syndromes are given by the product

$$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix},$$

which gives the standard decoding table with syndromes as

000	111	00
100	011	11
010	101	10
011	110	01

We will give a less trivial example in the next section, but first we need a better criterion for testing perfection.

For example, perfect codes $C \subset V(n, p)$ with $d = 3$ or 4 have $e = 1$ or 1.5. Thus every vector is of Hamming distance one from a unique codeword, that is differs from a unique codeword in at most one component. In particular, if C is also linear, there exists an obvious choice for the standard decoding table , the one for which the coset leaders lie in $B_e(\mathbf{0})$. But the elements of $B_e(\mathbf{0})$ of weight 1 are the $p-1$ nonzero multiples of the standard basis vectors of $V(n, p)$. Hence, for any standard decoding table for C with syndromes, the coset of any $\mathbf{v} \in V(n, p)$ can be immediately located by comparing its syndrome with the syndromes of the standard basis vectors and their multiples.

5.6.1 Testing for perfection

It turns out that there is a simple way of testing when a linear code is perfect. Let's first consider the binary case.

Proposition 5.16. *Suppose $C \subset V(n,2)$ is a linear code with $\dim C = k$. Then C is perfect if and only if $|B_e(\mathbf{0})| = 2^{(n-k)}$. Put another way, C is perfect if and only if*

$$\sum_{i=0}^{e} \binom{n}{i} = 2^{(n-k)}. \tag{5.4}$$

In particular, if $e = 1$, then C is perfect if and only if

$$(1+n)2^k = 2^n. \tag{5.5}$$

Proof. If C is perfect, then Theorem 5.15 tells us that $|B_e(\mathbf{0})| = 2^{n-k}$. But since a point in $B_e(\mathbf{0})$ has at most e nonzero components, we have

$$|B_e(\mathbf{0})| = \sum_{i=0}^{e} \binom{n}{i},$$

so (5.4) follows. Conversely, if (5.4) holds, then $|B_e(\mathbf{0})| = 2^{n-k}$, so there are 2^{n-k} distinct cosets $\mathbf{x} + C$ where $\mathbf{x} \in B_e(\mathbf{0})$. This implies every coset meets $B_e(\mathbf{0})$, so C is perfect. Applying this formula to the case $e = 1$ gives (5.5). □

Notice that $|B_e(\mathbf{0})|$ has nothing to do with C. The problem of finding a perfect code is to determine n and k so that $|B_e(\mathbf{0})| = 2^{(n-k)}$ and there exists a k-dimensional subspace C of $V(n,2)$ with $d(C) = 2e+1$ or $2e+2$. If $d(C) = 3$ or 4, then C is perfect if and only if $n = 2^{n-k} - 1$, where $k = \dim C$. Some possible solutions for these conditions are $n = 3$, $k = 1$ and $n = 7$, $k = 4$. We saw an example of the first case. The next example shows that a perfect code in the latter case (with $n = 7$ and $k = 4$) can be realized.

Example 5.13. Consider the 7-bit code C_7 defined as the row space of the matrix

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

By enumerating the 16 elements of C_7, one sees that $d(C_7) = 3$, so $e = 1$. Since $(7+1)2^4 = 2^7$, C_7 is perfect.

5.6.2 The hat problem

The hat problem is an example of an instance where the existence of a particular mathematical structure, in this case, perfect codes with $e = 1$ has a surprising application. Beginning with a simple case, let us describe the hat game. Suppose there are three players each wearing either a white hat or a black hat. Each player can see the hats of the other two players, but cannot see what color her own hat is. Furthermore the players are in sound proof booths and cannot communicate with each other. Each booth has three buttons marked B,W and P (for pass or no guess). At the sound of the buzzer, each player presses one of her three buttons. If nobody guesses incorrectly, and at least one player guesses correctly, then they share a $1,000,000 prize. The problem is this: assuming the players are allowed to formulate their strategy beforehand, how should they proceed to maximize their chances of winning?

Clearly, a pretty good strategy would be to have two players abstain and to have the third make a random guess. Their probability of winning with this strategy is a not bad 1/2. But this strategy doesn't make any use of fact that each player can see the hats of her two teamates. Suppose the following strategy is adopted: if a player sees that the other two players have the same colored hat, she guesses the opposite color. A player who sees different colored hats passes. With this strategy, the only losing hat configurations are BBB or WWW, so they win six out of eight times. Hence the probability of winning is at least a fantastic 3/4.

What does this have to do with perfect codes such that $e = 1$? If we represent Black by 0 and White by 1, then the various hat arrays are represented by the $2^3 = 8$ 3-bit strings. Let C be the 3-bit code $\{000, 111\}$. Thus C is a perfect code with $e = 1$. The above strategy amounts to the following. The three contestants agree ahead of time to assume that the hat configuration isn't in C. The probabilty of this happening is 3/4 since 6 out of 8 configurations aren't in C. Suppose that this assumption is correct. Then two players will see a 0 and a 1. They should automatically pass since there is no way of telling what their hat colors are. The third will see either two 0's or two 1's. If she sees two 0's, then (by assumption) she knows her hat is white and she hits the button marked W (for white). If she sees two 1's, then (by assumption) she knows her hat is black, and she hits the button marked B. This strategy fails only when the configuration lies in C.

Next, let's suppose there are 7 players imaginatively labelled $1, \ldots, 7$. If we still represent Black by 0 and White by 1, then the various hat arrays are represented by the 2^7 7-bit strings. Let's see if the strategy for three hats

still works with seven hats. First, all seven players need to memorize the 16 codewords of C_7. The players agree before the game starts to assume that the hat array isn't in C_7. Since $|C_7| = 2^4$, the probability that the hat array is in C_7 is $2^4/2^7 = 1/8$. Suppose (as for three hats) that their assumption is correct. Then the hat array $x_1 \dots x_7$ differs in one place from a codeword $c_1 \dots c_7$. Suppose this occurs at x_1. Then $x_2 = c_2, \dots, x_7 = c_7$. So player #1 sees $c_2 \dots c_7$ and recognizes that her hat color must be $c_1 + 1$ and guesses accordingly. Player #2 sees $x_1 c_3 \dots c_7$. But since $d(C_7) = 3$, she knows that whatever x_2 is, $x_1 x_2 c_3 \dots c_7 \notin C$. Therefore, she has to pass, as do the other five contestants. The chances that they win the million bucks are a pretty good 7/8.

Can you devise a strategy for how to proceed if there are 4,5 or 6 players? What about 8 or 9? More information about this problem and other related (and more serious) problems can be found in the article *The hat problem and Hamming codes* by M. Bernstein in Focus Magazine, November 2001.

Exercises

Exercise 5.26. Construct the parity check matrix and syndromes for C_7.

Exercise 5.27. Consider the code $C = \{00000, 11111\} \subset V(5, 2)$.

(i) Determine e.

(ii) Show that C is perfect.

Exercise 5.28. Show that any binary $[2^k - 1, 2^k - 1 - k]$-code with $d = 3$ is perfect. Notice C_7 is of this type.

Exercise 5.29. Can there exist a perfect code with $n = 5$ and $e = 2$?

Exercise 5.30. Suppose $n \equiv 2 \ mod(4)$. Show that there cannot be a1 perfect binary $[n, k]$-code with $e = 2$.

(iii) Does C present any possibilities for a five player hat game?

Exercise 5.31. Show that any binary $[23, 12]$-code with $d = 7$ is perfect.

5.7 Summary

Coding theory is the branch off mathematics that deals with error correcting codes. Coding theory is applied to design ways of transmitting data so that errors in transmission can be automatically corrected, as much as possible. A code is a subset C of some $V(n,p)$, and a linear code is a linear subspace. The elements of the code are called codewords. Linear coding theory provides an excellent (not to mention beautiul) realization of the principles we have already studied in elementary linear algebra: row operations, dimension, bases, cosets etc. The key concept is the notion of Hamming distance, which is an extremely simple natural metric on $V(n,p)$. The Hamming distance between two elements of $V(n,p)$ is the number of components where the two elements differ. The minimal distance of a code C is the least distance $d(C)$ between any two codewords. The main goal in coding theory is to devise codes C where $d(C)$ is large. This is due to the basic result on error correcting codes which says that an element $\mathbf{x}$ of $V(n,p)$ can have distance $d(\mathbf{x},\mathbf{c}) \leq e = \dfrac{1}{2}(d(C)-1)$ from at most one codeword $\mathbf{c}$. We discussed the codes which were used by the Mariner space probes in the 1970's. These codes are constructed from Hadamard matrices, namely $\{0,\pm 1\}$-matrices satisfying $HH^T = nI_n$. The main property of these codes is that e is very large.

Codes where every element of $V(n,p)$ is within $\dfrac{1}{2}(d(C)-1)$ units of a codeword are called perfect codes. A perfect code with $d(C)=3$ is called a Hamming code. These codes have the property that every element of $V(n,p)$ is either a codeword or one unit from a codeword. Infinitely many Hamming codes exist, but surprisingly, if we require that the minimum distance be greater than 3, there are only two examples of perfect codes. In fact, the question of whether or not a linear code is perfect comes down to solving the combinatorial condition (5.4).

A standard decoding table for a linear code C is a way of enumerating the cosets of C in $V(n,p)$. For example, if C is a Hamming code, then every coset contains a unique element of minimal weight (i.e. minimal Hamming distance from $\mathbf{0}$). This minimal element is assumed to be the error term for all elements in the coset. We also consider another application of linear algebra, namely the notion of the syndrome of a coset. For the final topic, we consider the *hat problem*, an amusing (and surprising) application of Hamming codes to the question of whether you know what color the hat you're wearing is.