Chapter 2
Error Correcting Codes

The identification number schemes we discussed in the previous chapter give us the ability to determine if an error has been made in recording or transmitting information. However, they are limited in two ways. First, the types of errors detected are fairly restrictive, e.g. single digit errors or interchanging digits. Second, they provide no way to recover the intended information. Some more sophisticated ideas and mathematical concepts enable methods to encoding and transmit information in ways that allow both detection and correction of errors. There are many applications of these so-called error correcting codes, among them transmission of digital images from planetary probes and playing compact discs and DVD movies.

2.1 Basic Notions

To discuss error correcting codes, we need first to set the context and define some terms. We work throughout in binary; that is, we will work over $\mathbb{Z}_2$. To simplify notation, we will write the two elements of $\mathbb{Z}_2$ as 0 and 1 instead of as $\overline{0}$ and $\overline{1}$. If n is a positive integer, then the set $\mathbb{Z}_2^n$ is the set of all n-tuples of $\mathbb{Z}_2$-entries. Elements of $\mathbb{Z}_2^n$ are called *words*, or words of length n. For convenience we will write elements of $\mathbb{Z}_2^n$ either with the usual notation, or as a concatenation of digits. For instance, we will write $(0, 1, 0, 1)$ and 0101 for the same 4-tuple. We can equip $\mathbb{Z}_2^n$ with an operation of addition by using point-wise addition. That is, we define

$$(a_1, \ldots, a_n) + (b_1, \ldots, b_n) = (a_1 + b_1, \ldots, a_n + b_n).$$

A consequence of the fact that $0 + 0 = 0 = 1 + 1$ in $\mathbb{Z}_2$ is that $a + a = \mathbf{0}$ for every $a \in \mathbb{Z}_2^n$, where $\mathbf{0}$ is the vector $(0, \ldots, 0)$ consisting of all zeros.

A *linear code* of length n is a nonempty subset of $\mathbb{Z}_2^n$ that is closed under the addition in $\mathbb{Z}_2^n$. Although nonlinear codes exist and are studied, linear codes are used most frequently in applications and much of the discussion simplifies greatly in this context. Because of their importance, we will consider only linear codes and drop the adjective "linear" from now on. We will refer to elements of a code as codewords.

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-04498-9_2) contains supplementary material, which is available to authorized users. The supplementary material can also be downloaded from http://extras.springer.com.

D.R. Finston and P.J. Morandi, *Abstract Algebra: Structure and Application*,
Springer Undergraduate Texts in Mathematics and Technology, DOI 10.1007/978-3-319-04498-9_2

Example 2.1. The set $\{00, 01, 10, 11\} = \mathbb{Z}_2^2$ is a code of length 2, and the set $\{0000, 1010, 0101, 1111\}$, which is a proper subset of $\mathbb{Z}_2^4$, is a code of length 4.

Let $w = a_1 \cdots a_n$ be a word of length n. Then the *weight* of w is the number of digits of w equal to 1. We denote the weight of w by $\mathrm{wt}(w)$. An equivalent and useful way to think about the weight of the word $w = a_1 \cdots a_n$ is to treat the a_i as the integers 0 or 1 (rather than as residue classes for the moment) and note that

$$wt(w) = \sum_{i=1}^{n} a_i .$$

There are some obvious consequences of this definition. First of all, $\mathrm{wt}(w) = 0$ if and only if $w = \mathbf{0}$. Second, $\mathrm{wt}(w)$ is a nonnegative integer. A more sophisticated fact about weight is its relation with addition. If $v, w \in \mathbb{Z}_2^n$, then $\mathrm{wt}(v+w) \leq \mathrm{wt}(v) + \mathrm{wt}(w)$. This is true because cancellation occurs when the ith components of v and w are both equal to 1. More precisely, write x_i for the ith component of a word x. The weight of x is then given by the equation $\mathrm{wt}(x) = |\{i : 1 \leq i \leq n, x_i = 1\}|$. Note that $(v + w)_i = v_i + w_i$, so that $(v + w)_i = 1$ implies that either $v_i = 1$ or $w_i = 1$ (but not both). Therefore,

$$\{i : 1 \leq i \leq n, (v + w)_i = 1\} \subseteq \{i : v_i = 1\} \cup \{i : w_i = 1\} .$$

Since $|A \cup B| \leq |A| + |B|$ for any two finite sets A, B, the inclusion above and the latter description of weight yields $\mathrm{wt}(v + w) \leq \mathrm{wt}(v) + \mathrm{wt}(w)$, as desired.

The idea of weight gives a notion of distance on $\mathbb{Z}_2^n$. If v, w are words, then we set the *distance* $D(v, w)$ between v and w to be

$$D(v, w) = \mathrm{wt}(v + w).$$

Alternatively, $D(v, w)$ is equal to the number of positions in which v and w differ. The function D shares the basic properties of distance in Euclidean space $\mathbb{R}^3$. More precisely, it satisfies the properties of the following lemma.

Lemma 2.2. *The distance function D defined on $\mathbb{Z}_2^n \times \mathbb{Z}_2^n$ satisfies:*

1. $D(v, v) = 0$ for all $v \in \mathbb{Z}_2^n$;
2. For any $v, w \in \mathbb{Z}_2^n$, if $D(v, w) = 0$, then $v = w$;
3. $D(v, w) = D(w, v)$ for any $v, w \in \mathbb{Z}_2^n$;
4. The Triangle Inequality: $D(v, w) \leq D(v, u) + D(u, w)$ for any $u, v, w \in \mathbb{Z}_2^n$.

Proof. Since $v + v = \mathbf{0}$, we have $D(v, v) = \mathrm{wt}(v + v) = \mathrm{wt}(\mathbf{0}) = 0$. This proves (1). We note that $\mathbf{0}$ is the only word of weight 0. Thus, if $D(v, w) = 0$, then $\mathrm{wt}(v + w) = 0$, which forces $v + w = \mathbf{0}$. However, adding w to both sides yields $v = w$, and this proves (2). The equality $D(v, w) = D(w, v)$ is obvious since $v + w = w + v$. Finally, we prove (4), the only non-obvious statement, with a cute argument. Given $u, v, w \in \mathbb{Z}_2^n$, we have, from the definition and the fact about the weight of a sum given above,

$$\begin{aligned} D(v, w) &= \mathrm{wt}(v + w) = \mathrm{wt}((v + u) + (u + w)) \\ &\leq \mathrm{wt}(v + u) + \mathrm{wt}(u + w) \\ &= D(v, u) + D(u, w). \end{aligned}$$

□

To discuss error correction we must first formalize the notion. Let C be a code. If w is a word, to correct, or decode, w means to select the codeword $v \in C$ such that

$$D(v, w) = \min\{D(u, w) : u \in C\}.$$

In other words, we decode w by choosing the closest codeword to w, under our notion of distance. There need not be a unique closest codeword, however. When this happens we can either randomly select a closest codeword, or do nothing. We refer to this notion of decoding as *maximum likelihood detection*, or MLD, the assumption being that the means of transmission of information is reliable so that if an error is introduced, the correct information is most likely to be the codeword that differs from the received word in the fewest number of positions.

Example 2.3. Let $C = \{00000, 10000, 011000, 11100\}$. If $w = 10001$, then w is distance 1 from 10000 and distance more than 1 from the other two codewords. Thus, we would decode w as 10000. However, if $u = 11000$, then u is distance 1 from both 10000 and from 111000. Thus, either is an appropriate choice to decode u.

We now define what it means for a code to be an error correcting code.

Definition 2.4. Let C be a code and let t be a positive integer. Then C is a t-error correcting code if whenever a word w differs from the nearest codeword v by a distance of at most t, then v is the unique closest codeword to w.

If a codeword v is transmitted and received as w, we can express w as $v + u$, and we say that $u = v + w$ is the error in transmission. As a word, the error u has a certain weight. So C is t-error correcting if for every codeword v and every word u whose weight is at most t, then v is the unique closest codeword to $v + u$.

If C is a t-error correcting code, then we say that C corrects t errors. Thus one way of interpreting the definition is that if v is a codeword, and if w is obtained from v by changing at most t entries of v, then v is the unique closest codeword to w. Therefore, by MLD decoding, w will be decoded as v.

Example 2.5. The code $C = \{000000, 111000, 000111\}$ is 1-error correcting. A word which differs from 000000 in one entry differs from the other two codewords in at least two entries. Similarly for the other two codewords in C.

Example 2.6. The code $C = \{00000, 10000, 011000, 11100\}$ above corrects no errors. Note that the word $u = 11000$ given in that example is a distance 1 from a codeword, but that codeword is not the unique closest codeword to u.

To determine for which t a code corrects t errors, we relate error correction to the distance of a code.

Definition 2.7. The distance d of a code is defined by

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

For intuitive purposes it may be useful to think of the minimum distance as the diameter of the smallest circle containing at least two codewords.

We denote by $\lfloor a \rfloor$ the greatest integer less than or equal to the number a.

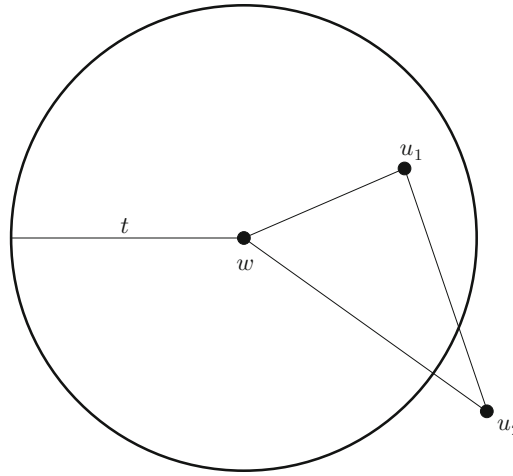
Proposition 2.8. *Let C be a code of distance d and set $t = \lfloor (d-1)/2 \rfloor$. Then C is a t-error correcting code but not a $(t+1)$-error correcting code.*

Proof. To prove that C is t-error correcting, let w be a word, and suppose that v is a codeword with $D(v, w) \leq t$. We need to prove that v is the unique closest codeword to w. We do this by

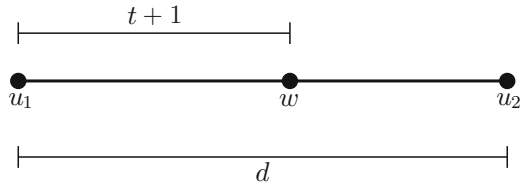
proving that $D(u,w) > t$ for any codeword $u \neq v$. If not, suppose that u is a codeword with $u \neq v$ and $D(u,w) \leq t$. Then, by the Triangle Inequality,

$$D(u,v) \leq D(u,w) + D(w,v) \leq t + t = 2t < d.$$

This is a contradiction to the definition of d. Thus v is indeed the unique closest codeword to w.



To finish the proof, we need to prove that C does not correct $t+1$ errors. Since the code has distance d, there are codewords u_1, u_2 with $d = D(u_1, u_2)$; in other words, u_1 and u_2 differ in exactly d positions. Let w be the word obtained from u_1 by changing exactly $t+1$ of those d positions. Then $D(u_1, w) = t+1$ and $D(u_2, w) = d-(t+1)$. Since $t = \lfloor (d-1)/2 \rfloor$ by our assumption, $(d-2)/2 \leq t \leq (d-1)/2$. In particular, $d-2 \leq 2t$ so that $D(u_2, w) = d-(t+1) \leq t+1$. Thus u_1 is not the unique closest codeword to w, since u_2 is either equally close or closer to w. Therefore C is not a $(t+1)$-error correcting code.



□

Example 2.9. Let $C = \{00000, 00111, 11100, 11011\}$. The distance of C is 3, and so C is a 1-error correcting code.

Example 2.10. Let n be an odd positive integer, and let $C = \{0\cdots0, 1\cdots1\}$ be a code of length n. If $n = 2t+1$, then C is a t-error correcting code since the distance of C is n. Thus, by making the length of C long enough, we can correct any number of errors that we wish. However, note that the fraction of components of a word that can be corrected is t/n, and this is always less than $1/2$.

Exercises

1. Find distance and error correction capability of the following codes:
   (a) $\{0000000, 1010101, 0101010, 1111111\}$,
   (b) $\{00000000, 11111111, 11100000, 00011111\}$,
   (c) $\{00000000,\ 11110000,\ 00001111,\ 10101010,\ 11111111,\ 01011010,\ 10100101,\ 01010101\}$.
2. Construct a linear code of length 5 with more than two codewords that corrects one error. Can you construct a linear code of length 4 with more than two words that corrects one error?
3. Let C be the code consisting of the solutions to the matrix equation $Ax = \mathbf{0}$, where

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

   Determine the codewords of C, and determine the distance and error correction capability of C.
4. Let A be a matrix, and let C be the code consisting of all solutions to $Ax = \mathbf{0}$. If A has neither a column of zeros nor two equal columns, prove that the distance of C is at least 3.
   (Hint: If v has weight 1 or weight 2, look at how Av can be written in terms of the columns of A.)
5. Let C be a code such that if $u, v \in C$, then $u + v \in C$. Prove that the distance of C is equal to the smallest weight of a nonzero codeword.
6. Let C be the code consisting of all solutions to a matrix equation $Ax = \mathbf{0}$. Let d be the largest integer such that any sum of fewer than d columns of A is nonzero. Prove that C has distance d.

2.2 Gaussian Elimination

In this section we recall some basic results about matrices, in particular Gaussian elimination, rank, and nullity. Our immediate concern is with matrices whose entries lie in $\mathbb{Z}_2$ in order to discuss the Hamming and Golay codes, historically the first examples of error correcting codes.

A system of linear equations is equivalent to a single matrix equation $AX = b$, where A is the matrix of coefficients, and X is the column matrix of variables. For example, the system of linear equations over the rational numbers

$$\begin{aligned} 2x + 3y - z &= 1 \\ x - y + 5z &= 2 \end{aligned}$$

is equivalent to the matrix equation

$$\begin{pmatrix} 2 & 3 & -1 \\ 1 & -1 & 5 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

The primary matrix-theoretic method for solving such a system is Gaussian elimination on the augmented matrix obtained from the coefficient matrix by appending on its right the column consisting of the right-hand side of the equation. Recall that Gaussian elimination employs operations on the rows

of a matrix, with the end result a matrix in row reduced echelon form. The latter represents a system of equations whose solutions, which are identical to those of the original system, can be found easily.

The three elementary row operations are :

- Replacing a row with a multiple of it by a nonzero scalar,
- Interchanging two rows,
- Replacing a row by its sum with a scalar multiple of another row.

In $\mathbb{Z}_2$ arithmetic the only multipliers available are 0 and 1 and $1 + 1 = 0$ in $\mathbb{Z}_2$ (so that $1 = -1$ and subtraction is the same operation as addition). In this context, the first of the three row operations listed above is not useful, since multiplying a row by 1 does not affect the row, and the third operation reduces to adding one row to another. The desired outcome is a matrix in row reduced echelon form:

Definition 2.11. A matrix A is in row reduced echelon form if all three of the following conditions are satisfied:

1. The first nonzero entry of each row is 1. This entry is called a leading 1.
2. If a column contains a leading 1, then all other entries of the column are 0.
3. If $i > j$, and if row i and row j each contain a leading 1, then the column containing the leading 1 of row i is further to the right than the column containing the leading 1 of row j.

To help understand Condition 3 of the definition, the leading 1's go to the right as you go from top to bottom in the matrix, so that the matrix is in some sense triangular.

Example 2.12. The following matrices over $\mathbb{Z}_2$ are in row reduced echelon form:

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

The columns with leading ones have the form of vectors e_i with a 1 in the ith position and 0's elsewhere.

In Chap. 4, familiar concepts from linear algebra over the real numbers will be systematically extended to include linear algebra over $\mathbb{Z}_2$. For now though, let's recall some facts about matrices with real entries in $\mathbb{R}$ that also hold for matrices with entries in $\mathbb{Z}_2$. First, the *row space* of a matrix is the vector space spanned by its rows. If the matrix is $m \times n$, then the rows are n-tuples, so the row space is a subspace of the space of all n-tuples. Since Gaussian elimination operates on the rows of a matrix in a reversible way, the row space of a matrix is identical with that of its row reduced echelon form. The *column space* of a matrix is the space spanned by the columns of the matrix. Again, if the matrix is $m \times n$, then the columns are m-tuples, so the column space is a subspace of the space of all m-tuples. These observations hold as well for matrices with entries in $\mathbb{Z}_2$. The only difference is that the span of a collection of rows or columns is merely the sum of some subset of them, again because the only multipliers available are 0 and 1.

The dimension of a vector space over $\mathbb{R}$ is the number of elements in a basis, provided this is finite. Otherwise the dimension is infinite. For an $m \times n$ matrix A, the dimension of the row space and the dimension of the column space are always finite and equal; this integer is called the *rank* of A. One benefit to reducing A to its row reduced echelon form E_A is that the nonzero rows of E_A (i.e., those that contain a leading 1) form a basis for the row space of A. Consequently, the dimension of the row space is the number of nonzero rows in E_A. Thus, an alternative definition of the rank of a matrix is the number of leading 1's in the row reduced echelon form obtained from the matrix. Again these assertions hold for matrices with entries in $\mathbb{Z}_2$.

The fact that the homogeneous linear systems $AX = 0$ and $E_A X = 0$ have the same solutions can be interpreted as the statement that the columns of A and the columns of E_A have the identical dependence relations (but their column spaces may be different). From Condition 2 it is clear that the columns of E_A that contain the leading 1's form a basis for its column space. Call these columns $c_{i1}, \ldots c_{ir}$. But then columns $i_1, \ldots i_r$ of the matrix A form a basis for its column space, hence the assertion above about the equality of the "row rank" and "column rank." It is clear also that the maximum possible rank of an $m \times n$ matrix is the minimum of m and n (although the matrix $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, for instance, shows that this bound need not be achieved).

Even though you might be most familiar with matrices whose entries are real numbers, the row operations above require only the ability to add, subtract, multiply, and divide the entries. In many situations, matrices arise whose entries are not real numbers, and our initial work in coding theory leads to matrices whose entries lie in $\mathbb{Z}_2$ (wherein we can certainly add, subtract, multiply, and divide, with the usual proscription against division by 0). Furthermore, all the theorems of linear algebra have analogues to this setting, and later on the fundamentals of linear algebra will be generalized to include other sets of scalars. Again, all that is necessary is closure of the scalars under the four arithmetic operations and the standard arithmetic properties analogous to those that hold for real number arithmetic (i.e., commutativity and associativity of addition and multiplication, and distributivity of multiplication over addition).

We now give several examples of reducing matrices with $\mathbb{Z}_2$ entries to echelon form. In each example once we have the matrix in row reduced echelon form, the leading 1's are marked in boldface.

Example 2.13. Consider the matrix

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

We reduce the matrix with the following steps. You should determine which row operation was done in each step.

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

The rank of A is equal to 3.

Example 2.14. Consider the matrix

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

To reduce this matrix, we can do the following steps.

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

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

The rank of A is equal to 3.

We now illustrate how the row reduced echelon form yields the solution of the systems of equations giving rise to the matrices in the previous examples.

Example 2.15. The system of equations

$$\begin{aligned} x &= 1\\ x+y &= 1\\ y+z &= 1 \end{aligned}$$

has augmented matrix

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

The reduction of this matrix

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

corresponds to the system of equations

$$\begin{aligned} x &= 1\\ y &= 0\\ z &= 1 \end{aligned}$$

and hence solves the original system.

Example 2.16. The augmented matrix

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

corresponds to the system of equations

$$\begin{aligned} x_1 + x_2 + x_5 &= 0 \\ x_1 + x_3 &= 1 \\ x_2 + x_3 + x_4 + x_5 &= 0 \\ x_2 + x_3 + x_5 &= 1. \end{aligned}$$

Reducing the matrix yields

$$\begin{pmatrix} \mathbf{1}\,0\,1\,0\,0\,1 \\ 0\,\mathbf{1}\,1\,0\,1\,1 \\ 0\,0\,0\,\mathbf{1}\,0\,1 \\ 0\,0\,0\,0\,0\,0 \end{pmatrix},$$

which corresponds to the system of equations

$$\begin{aligned} x_1 + x_3 &= 1 \\ x_2 + x_3 + x_5 &= 1 \\ x_4 &= 1. \end{aligned}$$

The leading 1's in boldface in the echelon matrix correspond to the variables x_1, x_2, and x_4. Solving for these yields the full solution

$$\begin{aligned} x_1 &= 1 + x_3, \\ x_2 &= 1 + x_3 + x_5 \\ x_4 &= 1 \\ & x_3 \text{ and } x_5 \text{ are arbitrary.} \end{aligned}$$

We can write out all solutions to this system of equations, since each of x_3 and x_5 can take on the two values 0 and 1. This gives us four solutions, which we write as row vectors:

$$(x_1, x_2, x_3, x_4, x_5) = (1 + x_3, 1 + x_3 + x_5, x_3, 1, x_5),$$

where $x_3 \in \{0, 1\}$ and $x_5 \in \{0, 1\}$.

The general solution is

$$(1 + x_3, 1 + x_3 + x_5, x_3, 1, x_5) = (1, 1, 0, 1, 0) + x_3(1, 1, 1, 0, 0) + x_5(0, 1, 0, 0, 1)$$

so that $(1, 1, 0, 1, 0)$, which corresponds to the values $x_3 = x_5 = 0$, yields a particular solution to the linear system. On the other hand, the vectors $(1, 1, 1, 0, 0), (0, 1, 0, 0, 1)$ solve the homogeneous system

$$\begin{aligned} x_1 + x_2 + x_5 &= 0, \\ x_1 + x_3 &= 0, \\ x_2 + x_3 + x_4 + x_5 &= 0, \\ x_2 + x_3 + x_5 &= 0. \end{aligned}$$

(Check this!) Thus any solution to the inhomogeneous system is obtained as the sum of a particular solution and a solution to the associated homogenous system.

Example 2.17. Let H be the *Hamming matrix* (named for Richard Hamming, mathematician, pioneer computer scientist, and inventor of the Hamming error correcting codes):

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

and consider the homogeneous system of equations $HX = \mathbf{0}$, where $\mathbf{0}$ refers to the 3×1 zero matrix and X is a 7×1 matrix of the variables $x_1, \ldots, x_7$. To solve this system we reduce the augmented matrix in one step to

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

yielding

$$\begin{pmatrix} \mathbf{1}\,0\,1\,0\,1\,0\,1\,0 \\ 0\,\mathbf{1}\,1\,0\,0\,1\,1\,0 \\ 0\,0\,0\,\mathbf{1}\,1\,1\,1\,0 \end{pmatrix}.$$

This matrix corresponds to the system of equations

$$\begin{aligned} x_1 + x_3 + x_5 + x_7 &= 0, \\ x_2 + x_3 + x_6 + x_7 &= 0, \\ x_4 + x_5 + x_6 + x_7 &= 0. \end{aligned}$$

Again, we have marked the leading 1's in boldface, and the corresponding variables can be solved in terms of the others, which can be arbitrary. So, the solution to this system is

$$\begin{aligned} x_1 &= x_3 + x_5 + x_7, \\ x_2 &= x_3 + x_6 + x_7, \\ x_4 &= x_5 + x_6 + x_7, \\ & x_3, x_5, x_6, x_7 \text{ are arbitrary.} \end{aligned}$$

Since we have four variables, x_3, x_5, x_6, and x_7, that can take on the values 0 or 1 in $\mathbb{Z}_2$ arbitrarily, there are exactly $2^4 = 16$ solutions to this system of equations.

To finish this chapter, we recall a theorem that will help us determine numeric data about error correcting codes. Before stating the theorem we explore the context in which it will be applied and recall some terminology.

The *kernel*, or *nullspace*, of a matrix A is the set of all solutions to the homogeneous equation $AX = \mathbf{0}$. As an illustration, consider the Hamming matrix H of the previous example.

Example 2.18. The solution above to the homogeneous equation $HX = \mathbf{0}$ can be described systematically by determining a basis for the nullspace of H. Since each distinct choice of the variables x_3, x_5, x_6, and x_7 in $\mathbb{Z}_2$ results in a unique solution to $HX = \mathbf{0}$, we obtain 4 solutions by successively setting one of these variables equal to 1 and all others arbitrary variables equal to 0, then using

$$\begin{aligned} x_1 &= x_3 + x_5 + x_7, \\ x_2 &= x_3 + x_6 + x_7, \\ x_4 &= x_5 + x_6 + x_7 \end{aligned}$$

to determine the values for the remaining variables. This technique results in the vectors

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

which form a basis for the nullspace of H. Indeed, the general solution of $HX = \mathbf{0}$ is given by

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = \begin{pmatrix} x_3 + x_5 + x_7 \\ x_3 + x_6 + x_7 \\ x_3 \\ x_5 + x_6 + x_7 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = x_3 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + x_5 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} + x_6 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} + x_7 \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix},$$

i.e., as a linear combination of the four specific solutions written above. A little work will show that every solution can be written in a unique way as a linear combination of these vectors. For example, check that $(0, 1, 1, 1, 1, 0, 0)$ is a solution to the system $HX = \mathbf{0}$. Writing this vector as a linear combination of the four given vectors, we must have $x_3 = x_5 = 1$ and $x_6 = x_7 = 0$, so

$$\begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

is a sum of two of the four given vectors, and can be written in no other way in terms of the four basis vectors.

This example indicates the general fact that for a homogeneous system $AX = \mathbf{0}$, the number of variables not corresponding to leading 1's (i.e., those above that could take on arbitrary values in $\mathbb{Z}_2$) is equal to the dimension of the nullspace of A. Let us call these variables *free variables* and the other variables (of which there are exactly the rank of A) *basic variables*. From the row reduced form of A, the basic variables can be expressed in terms of the free variables. Mimicking the example above, one obtains a distinguished set of solutions to $AX = 0$ by successively setting one free variable equal to 1 and the rest equal to 0. Then any solution can be written uniquely as a linear combination of these solutions. In particular this distinguished set of solutions is a basis for the nullspace of A and therefore, the number of free variables is equal to the dimension of the nullspace. Since every variable is either basic or free and the total number of variables is the number of columns of the matrix, we have the important rank-nullity theorem. The *nullity* of a matrix A is the dimension of the nullspace of A.

Theorem 2.19. *Let A be an $m \times n$ matrix. Then n is equal to the sum of the rank of A and the nullity of A.*

The point of this theorem is that once you know the rank of A, the nullity of A can be immediately calculated. Since we are working over $\mathbb{Z}_2$, the number of solutions to $AX = \mathbf{0}$ is then $2^{\text{nullity}(A)}$. In coding theory this will allow us to determine the number of codewords in a given code.

2.3 The Hamming Code

The Hamming code, discovered independently by Hamming and Golay, was the first example of an error correcting code. Let

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

be the Hamming matrix, described in Example 2.17 above. Note that the columns of this matrix give the base 2 representation of the integers 1–7. The Hamming code C of length 7 is the nullspace of H. More precisely,

$$C = \left\{ v \in \mathbb{Z}_2^7 : Hv^T = \mathbf{0} \right\}.$$

(The transpose is used here because codewords are typically written horizontally, i.e., as row vectors, but without commas to separate the entries). Just as the redundant check digit in an identification number enables the detection of certain errors by the failure of a certain dot product to result in 0, we will see that a code defined as the nullspaces of a matrix can introduce enough redundancies to enable the correction of certain errors.

Before proceeding to this topic, we use Gaussian elimination to gain more detailed information about the Hamming code. Solving as above the linear system $Hx = \mathbf{0}$, we obtain the solution

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = x_3 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + x_5 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} + x_6 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} + x_7 \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

Therefore, C has dimension 4, and the set $\{1110000, 1001100, 0101010, 1101001\}$ forms a basis for C (we will discuss these terms more rigorously in Chap. 4). If one were to write out all 16 codewords in C, one would find the distance of C to be exactly 3.

Linear codes like C are identified by their length, dimension, and minimum distance. Thus C is referred to as a $(7, 4, 3)$-code, because its length is 7, its dimension is 4, and its minimum distance is equal to 3. In particular, we deduce from Proposition 2.8 that C corrects 1 error.

The code C has a particularly elegant decoding algorithm, which we now describe. Let $\{e_1, \ldots, e_7\}$ be the standard basis for $\mathbb{Z}_2^7$. We point out a fact of matrix multiplication: He_i^T is equal to the ith column of H. Moreover, we note that the seven nonzero vectors in $\mathbb{Z}_2^3$ are exactly the seven columns of H.

Suppose that v is a codeword that is transmitted as a word $w \neq v$ and that exactly one error has been made in transmission. Then $w = v + e_i$ for some i. However, we do not yet know i, so we cannot yet determine v from w. However,

$$Hw^T = H(v + e_i)^T = Hv^T + He_i^T = He_i^T,$$

and He_i^T is the ith column of H, as we pointed out above. Therefore i is determined by computing Hw^T and comparing the result with the columns of H. The column number of H given by Hw^T is exactly i. Then w is decoded to $w + e_i$, which must be equal to v since we assumed that only one error was made in transmission. To summarize this error correcting algorithm: Given a word w, calculate Hw^T. If the product is 0, then w is a codeword. If it is not, then it is equal to the ith column of H for a unique integer i. Then $w + e_i$ is a valid codeword, and is the closest codeword to w.

The Hamming code C has an additional property: every word is within distance 1 of a codeword. To see this, suppose that w is a word. If $Hw^T = \mathbf{0}$, then w is a codeword. If not, then Hw^T is a nonzero 3-tuple. Therefore, it is equal to a column of H; say that Hw^T is equal to the ith column of H. Then $Hw^T = He_i^T$, so $H(w^T + e_i^T) = \mathbf{0}$, so that $w + e_i \in C$. The word $w + e_i$ is then a codeword a distance of 1 from w. A code that corrects t errors and for which every word is within t of some codeword is called *perfect*. Such codes are particularly nice, in part because a decoding procedure will always return a codeword. Later we will see some important codes that are not perfect. So perfection is not the ultimate goal. Nevertheless, we can be inspired by the words of Lord Chesterfield: "Aim at perfection in everything, though in most things it is unattainable. However, they who aim at it, and persevere, will come much nearer to it than those whose laziness and despondency make them give it up as unattainable."

Exercises

1. Let C be the code (of length n) of solutions to a matrix equation $Ax = \mathbf{0}$. Define a relation on the set $\mathbb{Z}_2^n$ of words of length n by $u \equiv v \bmod C$ if $u + v \in C$. Prove that this is an equivalence relation, and that for any word w, the equivalence class of w is the coset $C + w$.
2. Verify that 1100110 belongs to the $(7, 4, 3)$ Hamming code.
3. 1011110 is not a codeword for the $(7, 4, 3)$ Hamming code. Use the decoding algorithm above to identify the error and to correct it.
4. Consider the matrix $\hat{H}$ with entries in $\mathbb{Z}_2$ whose columns consist of the base 2 representations of the integers from 1 though 15 in increasing order. Determine the rank of $\hat{H}$ and find a basis for its nullspace.
5. Find the minimum distance and error correction capability of the nullspace of $\hat{H}$ defined in the previous problem. Is this code perfect?

2.4 Coset Decoding

To apply MLD (Maximum Likelihood Decoding, Sect. 2.1) what we must do, given a received word w, is search through all the codewords to find the codeword c closest to w. This can be a slow and tedious process. There are more efficient methods, assuming the code is built in a manner similar to that of the Hamming code, i.e., that the code C is given as the nullspace of an $m \times n$ matrix H:

$$C = \left\{v \in \mathbb{Z}_2^n : Hv^T = \mathbf{0}\right\}$$

and therefore has length n and dimension equal to the nullity of H. We fix the symbols C and H to have this meaning in this section.

Definition 2.20. Let w be a word. Then the coset $C + w$ of w is the set $\{c + w : c \in C\}$.

Recall two facts about C. First, by the definition of C, the zero vector $\mathbf{0}$ is an element of the code, since $H\mathbf{0} = \mathbf{0}$. From this we see that $w \in C + w$, since $w = \mathbf{0} + w$. Second, if $u, v \in C$, our assumption of linearity requires that $u + v \in C$ (i.e., $H(u + v)^T = Hu^T + Hv^T = \mathbf{0} + \mathbf{0} = \mathbf{0}$).

We now discuss an important property of cosets, namely that any two cosets are either equal or are disjoint. In fact cosets are the equivalence classes for the following equivalence relation defined on $\mathbb{Z}_2^n$:

$$\text{Two words } x \text{ and } y \text{ are related if } x + y \in C.$$

We write $x \sim y$ when this occurs. To see that this is an equivalence relation, we must verify the three properties of reflexivity, symmetry, and transitivity. For reflexivity, recall that addition in $\mathbb{Z}_2^n$ is componentwise so for every x in $\mathbb{Z}_2^n$ we have $x + x = \mathbf{0}$, which is an element of C. Thus $x \sim x$. Next, suppose that $x \sim y$. To verify symmetry, we must show that $y \sim x$. The assumption that $x \sim y$ means $x + y \in C$. However, $x + y = y + x$; therefore, since $y + x \in C$, we have $y \sim x$. Finally, for transitivity, suppose that $x \sim y$ and $y \sim z$. Then $x + y \in C$ and $y + z \in C$. Adding these codewords results in a codeword by the previous paragraph. However,

$$(x + y) + (y + z) = x + (y + y) + z = x + \mathbf{0} + z = x + z,$$

by the properties of vector addition. Since the result, $x + z$, is an element of C, we have $x \sim z$, as desired. So ~ is an equivalence relation.

The equivalence class of a word x is

$$\begin{aligned}\{y : y \sim x\} &= \{y : x + y \in C\} = \{y : y = c + x \text{ for some } c \in C\} \\ &= C + x.\end{aligned}$$

The third equality follows since if $x + y = c$, then $y = c + x$.

Proposition 2.21. *If x and y are words, then $C + x = C + y$ if and only if $Hx^T = Hy^T$.*

Proof. Suppose first that $C + x = C + y$. Then $x \sim y$, so $x + y \in C$. By definition of C, we have $H(x + y)^T = \mathbf{0}$. Expanding the left-hand side, and using the fact that $(x + y)^T = x^T + y^T$, we get $Hx^T + Hy^T = \mathbf{0}$, so $Hx^T = Hy^T$. Conversely, suppose that $Hx^T = Hy^T$. Then

$Hx^T + Hy^T = \mathbf{0}$, or $H(x+y)^T = \mathbf{0}$. This last equation says $x + y \in C$, and so $x \sim y$. From this relation between x and y, we obtain $C + x = C + y$, since these are the equivalence classes of x and y, and these classes are equal since x and y are related. □

Example 2.22. Let

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

A short calculation shows that $C = \{0000, 1100, 0011, 1111\}$. The cosets of C are then seen to be

$$\begin{aligned} C + 0000 &= \{0000, 1100, 0011, 1111\}, \\ C + 1000 &= \{1000, 0100, 1011, 0111\}, \\ C + 0010 &= \{0010, 1110, 0001, 1101\}, \\ C + 1010 &= \{1010, 0110, 1001, 0101\}. \end{aligned}$$

We also point out that $C = C + 0000 = C + 1100 = C + 0011 = C + 1111$; in other words, $C = C + v$ for any $v \in C$. Each coset in this example is equal to the coset of four vectors, namely the four vectors in the coset.

Introducing some coding theory terminology, call Hx^T the *syndrome* of x. Syndromes enable more efficient decoding. Suppose that a word w is received. If c is the closest codeword to w, let $e = c + w$. Then e is the *error word*, in that e has a digit equal to 1 exactly when that digit was transmitted incorrectly in c. Note that e is the word of smallest possible weight of the form $v + w$ with $v \in C$ since $\mathrm{wt}(e) = D(c, w)$. If we can determine e, then we can determine c by $c = e + w$. To see where the syndrome comes into play, multiply both sides of the equation $e^T = c^T + w^T$ by H to obtain

$$\begin{aligned} H^e T &= H(c+w)^T = Hc^T + Hw^T = \mathbf{0} + Hw^T \\ &= Hw^T \end{aligned}$$

which is the syndrome of the received word. We therefore compute He^T by computing Hw^T. Proposition 2.21 says that $C + e = C + w$; in other words, $e \in C + w$. More generally, any pair of words with the same syndrome determine the same coset of C. Since c is the closest codeword to w, the word e is then the word of least weight in the coset $C + w$. We then find e by searching the words in $C + w$ for the word of least weight; such a word is called a *coset leader*. To decode with cosets, we compute and list a coset leader for each coset (i.e., syndrome).

Example 2.23. Let

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

Then $C = \{00000, 11100, 00111, 11011\}$. We see that the distance of C is 3, so C is 1-error correcting. The cosets of C are

$$\{\mathbf{00000}, 00111, 11011, 11100\},$$
$$\{01110, \mathbf{10010}, 01001, 10101\},$$
$$\{\mathbf{00010}, 00101, 11001, 11110\},$$
$$\{11111, 11000, 00011, \mathbf{00100}\},$$
$$\{01111, \mathbf{01000}, 10100, 10011\},$$
$$\{01101, 10110, \mathbf{01010}, 10001\},$$
$$\{01100, \mathbf{10000}, 10111, 01011\},$$
$$\{11010, \mathbf{00001}, 11101, 00110\}.$$

By searching through each of the eight cosets (a word of minimal weight in each coset has been boldfaced), we can then build the following syndrome table:

Syndrome	**Coset leader**
000	00000
101	10010
010	00010
011	00100
100	01000
110	01010
111	10000
001	00001

The following examples illustrate the use of a syndrome table for decoding. Suppose that $w = 10010$ is received. Calculating $(Hw^T)^T = wH^T$ results in 101. First of all, since $Hw^T \neq \mathbf{0}$, and by the definition of the code as the nullspace of H, the vector w is not a codeword. From the syndrome table, we see that 101 is the second syndrome listed. The corresponding coset leader is $e = 10010$. The received word w is decoded as $c = w + e = 00000$. Similarly, if we receive the word $w = 11111$, we calculate $wH^T = 011$. The corresponding coset leader is $e = 00100$, so the corrected codeword is $e + w = 11011$.

Clearly using the syndrome table requires much less computation than checking the distance between w and all 16 codewords to find the closest one. The fact that choices of the weight 2 coset leader were made for syndromes 110 and 101 shows that this code cannot correct two errors and also that it is not perfect.

Exercises

1. Let C be the code consisting of all solutions of the matrix equation $Ax^T = \mathbf{0}$, where

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

a. Calculate C and determine its distance and error correcting capability.
b. Construct the syndrome table for C.
c. Use the table to decode the vectors 10101101, 01011011, and 11000000.

2. List all of the cosets of the code $C = \{00000, 11100, 00111, 11011\}$.
3. Find the cosets of the Hamming code.
4. Let C be the code consisting of solutions to $Ax^T = \mathbf{0}$, where

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

Build the syndrome table for C. Determine the distance of C. Use it to decode, if possible, 111110 and 100000. Feel free to use the Maple worksheet Cosets.mw.

2.5 The Golay Code

In this section we discuss a length 24 code used by NASA in the 1970s and 1980s to transmit images of Jupiter and Saturn photographed by the Voyager spacecraft. This code, called the *extended Golay code,* is the set of solutions to the matrix equation $Hx^T = \mathbf{0}$, where H is the 12×24 matrix $H = [I \mid B]$ whose left half is the 12×12 identity matrix I and whose right half is the symmetric 12×12 matrix

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

which satisfies $B^2 = I$.

The photographs were made using 4,096 colors. Each color was encoded with a codeword from the Golay code. By solving the matrix equation $Hx^T = \mathbf{0}$, we can see that there are indeed 4,096 codewords. Furthermore, a tedious check of all codewords shows that the distance of the Golay code has distance $d = 8$. Thus, the code can correct $\lfloor (8-1)/3 \rfloor = 3$ errors, hence up to three out of the 24 digits of a codeword can be corrupted and still the original information will be retrievable.

Because this code can correct more than one error, any decoding procedure is bound to be more complicated than that for the Hamming code. We give a decoding procedure based on some simple facts about the matrix B. Its validity is left to a series of homework problems.

To make it more convenient to work with this code, we write a word $u = (u_1, u_2)$, where u_1 consists of the first 12 digits and u_2 the remaining 12. Since $H = [I \mid B]$, we see that $u \in C$ if and only if

$Hu^T = \mathbf{0}$, which is true if and only if $u_1^T + Bu_2^T = \mathbf{0}$. For a received word w, the following steps are performed to decode w. We write v for the codeword to be determined from w. As usual, e_i denotes the 12-tuple with ith-entry 1 and all other entries 0, while b_i denotes the ith row of the matrix B.

1. Compute $s^T = Hw^T$. If $s^T = \mathbf{0}$, then w is a codeword.
2. If $1 \leq \text{wt}(s) \leq 3$, then $v = w + (s, \mathbf{0})$.
3. If $\text{wt}(s) > 3$ and $\text{wt}(s + b_i) \leq 2$ for some i, then $v = w + (s + b_i, e_i)$.
4. If we haven't yet determined v, then compute sB, which is equal to $(Bs^T)^T$ by symmetry of B.
5. If $1 \leq \text{wt}(sB) \leq 3$, then $v = w + (0, sB)$.
6. If $\text{wt}(sB) > 3$ and $wt(sB + b_i) \leq 2$ for some i, then $v = w + (e_i, sB + b_i)$.
7. If we haven't determined v, then w cannot be decoded.

Example 2.24. Suppose that $w = 001001001101101000101000$ is received. We calculate $s^T = Hw^T$, and find $s = 110001001001$ with $\text{wt}(s) = 5$. We see that $\text{wt}(s + b_5) = 2$. Therefore, by Step 3, w is decoded as $v = w + (s + b_5, e_5) = w + (000000010010, 000010000000) = 001001011111101010101000$.

Exercises

For these problems, some of the theoretical facts behind the decoding procedure for the Golay code are verified. We use the following setup: C is the Golay code, H is the 12×24 matrix $[I \mid B]$ mentioned in the text, w is a received word, $s^T = Hw^T$. Our conventions are that a 24-tuple written as (u_1, u_2) means that each u_i is a 12-tuple and that the ith row (and column) of the symmetric matrix B is denoted by b_i. Let v be the closest codeword to w and write $v = w + e$. Since the Golay code is asserted to be 3-error correcting, we assume that $\text{wt}(e) \leq 3$.

Recall that $B^2 = I$ and $B^T = B$. A straightforward but tedious check of the rows of B shows that (i) $\text{wt}(b_i) \geq 7$ for all i; (ii) $\text{wt}(b_i + b_j) \geq 6$ if $i \neq j$; (iii) $\text{wt}(b_i + b_j + b_k) \geq 5$ for all i, j, k. Since $B^T = B$, the ith column of B is b_i, and so $Be_i = b_i$. You are free to use these facts.

1. Suppose that $e = (u, \mathbf{0})$; with $\text{wt}(u) \leq 3$. Show that $s = u$, and conclude that $v = w + (s, \mathbf{0})$.
2. Suppose that $e = (u, e_i)$ with $\text{wt}(u) \leq 2$. Show that $s = u + b_i$. Conclude that $\text{wt}(s) > 3$ and $\text{wt}(s + b_i) \leq 2$, and that $v = w + (s + b_i, e_i)$.
3. Suppose that $e = (\mathbf{0}, u)$ with $\text{wt}(u) \leq 3$. Show that s is the sum of at most three of the b_i and that $u = sB$. Conclude that $\text{wt}(s) > 3$ but $\text{wt}(Bs) \leq 3$, and that $v = w + (\mathbf{0}, sB)$.
4. Suppose that $e = (e_i, u)$ with $\text{wt}(u) \leq 2$. Show that $s = e_i + uB$, and that $sB = b_i + u$. Conclude that $\text{wt}(s) > 3$, $\text{wt}(s + b_i) > 2$ for any i, and that $e = (e_i, sB + b_i)$, so $v = w + (e_i, sB + b_i)$.

These four problems show, given any possibility of an error vector e having weight at most 3, how we can determine it in terms of the syndrome s. Reading these four problems backwards yields the decoding procedure discussed in this section.

References

1. Hankerson DC et al (2000) Coding theory and cryptography: the essentials, 2nd edn. Marcel Dekker, New York
2. Herstein I (1975) Topics in algebra, 2nd edn. Wiley, Hoboken
3. Talbot J, Welsh D (2006) Complexity and cryptography: an introduction. Cambridge University Press, Cambridge

http://www.springer.com/978-3-319-04497-2

Abstract Algebra
Structure and Application
Finston, D.; Morandi, P.
2014, IX, 187 p. 45 illus. With online files/update.,
Hardcover
ISBN: 978-3-319-04497-2
A product of Birkhäuser Basel