Chapter 3

Linear Codes

In order to define codes that we can encode and decode efficiently, we add more structure to the codespace. We shall be mainly interested in linear codes. A *linear code* of length n over the field F is a subspace of F^n. Thus the words of the codespace F^n are vectors, and we often refer to codewords as *codevectors*.

linear code

codevectors

In the first section we develop the basics of linear codes, in particular we introduce the crucial concept of the dual of a code. The second and third sections then discuss the general principles behind encoding and decoding linear codes. We encounter the important concept of a syndrome.

3.1 Basics

If C is a linear code that, as a vector space over the field F, has dimension k, then we say that C is an $[n, k]$ *linear code* over F, or an $[n, k]$ code, for short. There is no conflict with our definition of the dimension of C as a code, since $|C| = |F|^k$. (Indeed the choice of general terminology was motivated by the special case of linear codes.) In particular the rate of an $[n, k]$ linear code is k/n. If C has minimum distance d, then C is an $[n, k, d]$ linear code over F. The number $n - k$ is again the *redundancy* of C.

$[n, k]$ linear code

redundancy

We begin to use $\mathbb{F}_2$ in preference to $\{0, 1\}$ to denote our binary alphabet, since we wish to emphasize that the alphabet carries with it an arithmetic structure. Similar remarks apply to ternary codes.

EXAMPLES. (i) The repetition code of length n over F is an $[n, 1, n]$ linear code.

(ii) The binary parity check code of length n is an $[n, n-1, 2]$ linear code.

(iii) The $[7, 4]$, $[8, 4]$, and $[4, 2]$ Hamming codes of the introduction were all defined by parity considerations or similar equations. We shall see below that this forces them to be linear.

(iv) The real Reed-Solomon code of our example is a $[27, 7, 21]$ linear code over the real numbers $\mathbb{R}$.

(3.1.1) THEOREM. (SHANNON'S THEOREM FOR LINEAR CODES.) *Let F be a field with m elements, and consider a $mSC(p)$ with $p < 1/m$. Set*

$$\mathcal{L}_\kappa = \{ \text{ linear codes over } F \text{ with rate at least } \kappa \}.$$

Then $\mathcal{L}_\kappa$ is a Shannon family provided $\kappa < C_m(p)$. □

Forney (1966) proved a strong version of this theorem which says that we need only consider those linear codes of length n with encoder/decoder complexity on the order of n^4 (but at the expense of using very long codes). Thus there are Shannon families whose members have rate approaching capacity and are, in a theoretical sense, practical[1].

Hamming weight

The *Hamming weight* (for short, *weight*) of a vector $\mathbf{v}$ is the number of its nonzero entries and is denoted $\mathrm{w_H}(\mathbf{v})$. We have $\mathrm{w_H}(\mathbf{x}) = \mathrm{d_H}(\mathbf{x}, \mathbf{0})$. The *minimum weight* of the code C is the minimum nonzero weight among all codewords of C,

minimum weight

$$\mathrm{w}_{\min}(C) = \min_{\mathbf{0} \neq \mathbf{x} \in C} (\mathrm{w_H}(\mathbf{x})) .$$

(3.1.2) LEMMA. *Over a field, Hamming distance is translation invariant. In particular, for linear codes, the minimum weight equals the minimum distance.*

PROOF. Clearly $\mathrm{d_H}(\mathbf{x}, \mathbf{y}) = \mathrm{d_H}(\mathbf{x} - \mathbf{z}, \mathbf{y} - \mathbf{z})$ for all $\mathbf{z}$. In particular

$$\mathrm{d_H}(\mathbf{x}, \mathbf{y}) = \mathrm{d_H}(\mathbf{x} - \mathbf{y}, \mathbf{y} - \mathbf{y}) = \mathrm{d_H}(\mathbf{x} - \mathbf{y}, \mathbf{0}) . \qquad \square$$

A consequence of the lemma is that minimum distance for linear codes is much easier to calculate than for arbitrary codes. One need only survey $|C|$ codewords for weight rather than roughly $|C|^2$ pairs for distance.

EXAMPLES. Of course the minimum weight of the length n repetition code is n. Also the minimum weight of the parity check code is clearly 2. The minimum weight of the length 27 real Reed-Solomon code is equal to its minimum distance which we found to be 21. We listed the codewords of the $[4, 2]$ ternary Hamming code, and so it visibly has minimum weight 3.

Verifying that the minimum weight of the $[7, 4]$ Hamming code is 3 is easy to do directly by hand, but we will give a conceptual way of doing this calculation below. The extended $[8, 4]$ Hamming code adds an overall parity check bit to the $[7, 4]$ code, so its minimum weight is 4.

The following elementary property of binary weights can be very helpful. For instance, it proves directly that the parity check code is linear.

(3.1.3) PROBLEM. *Prove that, for binary vectors $\mathbf{x}$ and $\mathbf{y}$ of the same length, we have*

$$\mathrm{w_H}(\mathbf{x} + \mathbf{y}) = \mathrm{w_H}(\mathbf{x}) + \mathrm{w_H}(\mathbf{y}) - 2\mathrm{w_H}(\mathbf{x} * \mathbf{y})$$

where $\mathbf{x} * \mathbf{y}$ is defined to have a 1 only in those positions where both $\mathbf{x}$ and $\mathbf{y}$ have a 1.

The matrix G is a *spanning matrix* for the linear code C provided $C = \mathrm{RS}(G)$, the row space of G. A *generator matrix* of the $[n,k]$ linear code C over F is a $k \times n$ matrix G with $C = \mathrm{RS}(G)$. Thus a generator matrix is a spanning matrix whose rows are linearly independent. We may easily construct many codes using generator matrices. Of course it is not clear from the matrix how good the code will be.

spanning matrix
generator matrix

EXAMPLES. (i) The repetition code has generator matrix

$$G = \Big[1, 1, \ldots, 1\Big] .$$

(ii) A particularly nice generator matrix for the parity check code is

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 1 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & 1 & \cdots & 0 & 0 & 1 \\ & \vdots & & \ddots & & \vdots & \\ 0 & 0 & 0 & \cdots & 1 & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 1 \end{bmatrix} ,$$

composed of all weight 2 codewords with a one in the last column. This code will have many other generator matrices as well. Here are two for the $[7,6]$ parity check code:

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} , \quad \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} .$$

(iii) Consider the $[7,4]$ Hamming code of Example 1.3.3. In turn we set the four message symbols (X_3, X_5, X_6, X_7) to $(1,0,0,0)$, $(0,1,0,0)$, $(0,0,1,0)$, and $(0,0,0,1)$. The four resulting codewords form the rows of a generator matrix. We find

$$\begin{bmatrix} 1 & 1 & \mathbf{1} & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 1 & 0 & \mathbf{0} & 1 & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ 0 & 1 & \mathbf{0} & 1 & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ 1 & 1 & \mathbf{0} & 1 & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

(iv) A generator matrix for the $[8,4]$ extended Hamming code of Example 1.3.4 results from adding a column at the front to that for the $[7,4]$ code, each new entry checking parity of that row in the matrix. We have

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

[1] Oxymoron!

(v) For a generator matrix of the $[4, 2]$ ternary Hamming code of Example 1.3.5, we may set (a, b) equal to $(1, 0)$ and $(0, 1)$ in turn to get the matrix

$$\begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 1 \end{bmatrix},$$

although any pair of codewords would do as rows provided one is not a multiple of the other. For instance

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 2 & 0 \end{bmatrix}$$

is also a generator matrix.

(3.1.4) PROBLEM. *Prove that, in a linear code over the field $\mathbb{F}_q$, either all of the codewords begin with 0 or exactly $1/q$ of the codewords begin with 0. (You might want first to consider the binary case.)*

(3.1.5) PROBLEM. *Let C be an $[n, k, d]$ linear code over the field $\mathbb{F}_q$.*

*(a) Prove that the sum of all the weights of all the codewords of C is at most $n(q-1)q^{k-1}$. (*HINT: *Use the previous problem.)*

*(b) Prove that the minimum distance d of C is at most $\dfrac{n(q-1)q^{k-1}}{q^k - 1}$. (*HINT: *The minimum weight is less than or equal to the average nonzero weight.)*

(c) Prove the Plotkin bound for linear codes with $d/n > (q-1)/q$:

$$|C| \leq \frac{d}{d - \frac{q-1}{q}n} .$$

(3.1.6) PROBLEM. *Prove the Plotkin bound for a general m-ary code C of length n and minimum distance d with $d/n > (m-1)/m$:*

$$|C| \leq \frac{d}{d - \frac{m-1}{m}n} .$$

*(*HINT: *Find an upper bound on the average nonzero distance between codewords by comparing all distinct pairs of codewords and examining each coordinate position in turn.)*

dual code

Let C be any code (not necessarily linear) in F^n, for F a field. The *dual code* of C, denoted $C^\perp$, is the code

$$C^\perp = \{\mathbf{x} \in F^n \mid \mathbf{x} \cdot \mathbf{c} = 0, \text{ for all } \mathbf{c} \in C\},$$

where $\mathbf{x} \cdot \mathbf{c}$ is the usual dot product. The dual of C is linear even if C is not. (This is often a good way of proving that a given code is linear.) We can in turn examine the dual of the dual and discover easily that $(C^\perp)^\perp = C^{\perp\perp} \supseteq C$.

If C is itself a linear code, then in fact $C^{\perp\perp} = C$. For instance, the dual of the binary repetition code of length n is the parity check code of length n; and the dual of the parity check code of length n is the repetition code of length n. To see that $C^{\perp\perp} = C$ for linear C, we use another description of $C^\perp$. Let G be a generator matrix for C. Then $\mathbf{x}$ is in $C^\perp$ if and only if $G\mathbf{x}^\top = \mathbf{0}$. Thus

the vectors of $C^\perp$ are precisely the transposes of the vectors of the null space $\mathrm{NS}(G)$. Therefore by Theorem A.1.7 the dimension of C plus the dimension of $C^\perp$ equals the length n, that is, $C^\perp$ has dimension $n-k$. Calculating dimensions twice, we learn that $C^{\perp\perp}$ has dimension k. As this space contains C and has the same dimension as C, it is equal to C. In summary:

(3.1.7) LEMMA. *If C is an $[n,k]$ linear code over F, then its dual $C^\perp$ is an $[n,n-k]$ linear code over F and $C^{\perp\perp}=C$.* □

The linear code C is *self-orthogonal* if $C^\perp \geq C$ and is *self-dual* if $C^\perp = C$. So, for instance, a binary repetition code of even length is self-orthogonal, as is the $[7,3]$ binary dual Hamming code. Since the dimension of a code plus that of its dual add up to the length, a self-dual code must be a $[2k,k]$ linear code, for some k. The $[8,4]$ extended Hamming code is self-dual, as can be easily checked using the generator matrix given above. The ternary $[4,2]$ Hamming code is also self-dual, as is easily checked.

self-orthogonal

self-dual

A generator matrix H for the dual code $C^\perp$ of the linear C is sometimes called a *check matrix* for C. In general it is not difficult to calculate a check matrix for a code, given a generator matrix G. Indeed if we pass to a generator in **RREF**, then it is easy to find a basis for the null space and so for $C^\perp$ by following the remarks of Section A.1.3 of the appendix. In particular, if the generator matrix G (or its **RREF**) has the special form

check matrix

$$\left[\; I_{k\times k} \mid A_{k\times n-k} \;\right]$$

then one check matrix is

$$H = \left[-A^\top_{n-k\times k} \mid I_{n-k\times n-k}\;\right].$$

(3.1.8) PROBLEM. *Consider a binary code of length* 16 *written as* 4×4 *square matrices. The code E is composed of every 4×4 binary matrix M such that:*

(*i*) *every row of M contains an even number of* 1*'s; and*

(*ii*) either *every column of M contains an even number of* 1*'s* or *every column of M contains an odd number of* 1*'s.*

(*a*) *Prove that E is a linear code.*

(*b*) *What is the dimension of E?*

(*c*) *What is the minimum distance of E?*

(*d*) *If the matrix*

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

is received, give all possible decodings subject to **MDD**. *That is, find all code matrices in E that are at minimum distance from this matrix.*

(3.1.9) PROBLEM. *Consider a binary code of length* 21 *whose words are written as arrays in the following gem shape:*

$$\begin{array}{ccccc} & x_1 & x_2 & x_3 & \\ x_4 & x_5 & x_6 & x_7 & x_8 \\ x_9 & x_{10} & x_{11} & x_{12} & x_{13} \\ x_{14} & x_{15} & x_{16} & x_{17} & x_{18} \\ & x_{19} & x_{20} & x_{21} & \end{array}$$

The code E is composed of every binary array M of this shape and such that:
(i) every row of M contains an even number of 1*'s; and*
(ii) every column of M contains an even number of 1*'s.*

(a) Prove that E is a linear code.
(b) What is the dimension of E?
(c) What is the minimum distance of E?
(d) If the array

$$\begin{array}{ccccc} & 0 & 1 & 0 & \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ & 0 & 1 & 0 & \end{array}$$

is received, give all possible decodings subject to **MDD**. *That is, find all codewords in E that are closest to this array.*

(e) If the array

$$\begin{array}{ccccc} & 1 & 0 & 1 & \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ & 1 & 0 & 1 & \end{array}$$

is received, give all possible decodings subject to **MDD**.

(3.1.10) PROBLEM. *If C is a binary $[n, k]$ linear code, prove that either all weights of codewords of C are even or the even weight codewords of C form a linear $[n, k-1]$ subcode B. In the second case, how can the dual code of B be constructed from the dual code of C?*

(3.1.11) PROBLEM. *(a) Let C be a self-orthogonal binary linear code. Prove that all of its codewords have even weight. If additionally C has a spanning set composed of codewords with weights a multiple of* 4*, prove that every codeword has weight a multiple of* 4*.*

(b) Prove that a linear ternary code is self-orthogonal if and only if all its weights are a multiple of three.

If C is a binary code and $\mathbf{x}$ is a vector of $C^{\perp}$ then $\mathbf{c} \cdot \mathbf{x} = 0$, for all $\mathbf{c} \in C$; so $\mathbf{x}$ can be thought of as checking the parity of a subset of the coordinate positions of C, those positions in which $\mathbf{x}$ equals one. Extending this idea to nonbinary linear codes, we consider any vector of the dual as providing the coefficients of a "parity check equation" on the entries of codewords. The rows of a check matrix provide a basis for the space of parity check equations satisfied by the code, hence the terminology.

Because $C^{\perp\perp} = C$, we can use a check matrix H for C to give a concise definition of C:

$$C = \{\, \mathbf{x} \mid H\mathbf{x}^\top = \mathbf{0} \,\}.$$

Any matrix H for which $C = \{\, \mathbf{x} \mid H\mathbf{x}^\top = \mathbf{0} \,\}$ we shall call a *control matrix* for C. (This terminology is not common.) Thus a check matrix is a special kind of control matrix. A check matrix must have linearly independent rows while a control matrix need not.

control matrix

We often define a code in terms of a check matrix (or control matrix). In Example 1.3.5 we defined the $[4, 2]$ ternary Hamming code to be all 4-tuples (a, b, c, d) from $\{0, 1, 2\}^4$ that satisfy $a + b = c$ and $b + c + d = 0$. That is, we defined the code via the check matrix

$$\begin{bmatrix} 1 & 1 & 2 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}.$$

Here the first check row requires that, for (a, b, c, d) to be in the code,

$$(a, b, c, d) \cdot (1, 1, 2, 0) = a + b + 2c = 0\,,$$

that is, $a + b = c$; and the second forces $b + c + d = 0$.

Shannon's discussion under Examples 1.3.3 of the $[7, 4]$ binary Hamming code essentially defines the code by its check matrix

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

Here every nonzero binary 3-tuple occurs exactly once as a column of the check matrix. The columns have been arranged so that column i is the binary representation of the integer i.

What is the minimum weight of the $[7, 4]$ Hamming code? If $\mathbf{x}$ is a vector of weight 1, then the product $H\mathbf{x}^\top$ is a column of H, indeed column i of H if the single 1 of $\mathbf{x}$ is in position i. As all columns of H are nonzero, $H\mathbf{x}^\top$ is also nonzero; so $\mathbf{x}$ is not a codeword. If instead $\mathbf{x}$ has weight 2, then $H\mathbf{x}^\top$ is the sum of two columns of H, those columns in which $\mathbf{x}$ equals 1. As no two columns are equal, this sum is never $\mathbf{0}$; so again $\mathbf{x}$ is not a codeword. On the other hand, it is possible to find three columns of H that sum to $\mathbf{0}$ (for instance, the first three); so the code does contain words of weight 3 (for instance, $(1, 1, 1, 0, 0, 0, 0)$). Therefore this code has minimum weight 3.

It is not difficult to generalize these arguments. Block matrix multiplication implies that, for any matrix H and row vector $\mathbf{x}$, the matrix product $H\mathbf{x}^\top$ is a linear combination of the columns of H with coefficients provided by $\mathbf{x}$, namely the sum $\sum_i \mathbf{h}_i x_i$ where H has i^{th} column $\mathbf{h}_i$ and x_i is the i^{th} entry of $\mathbf{x}$. In particular the entries of a nonzero codeword $\mathbf{x}$ give the coefficients of a linear dependence among the columns of H, a check matrix (or control matrix). Of course any column $\mathbf{h}_i$ that is multiplied by a scalar $x_i = 0$ makes no contribution to this linear combination. The nonzero entries of the codeword are the coefficients of a linear dependence among only those columns $\mathbf{h}_i$ for which the coefficient x_i is not 0. We are led to:

(3.1.12) LEMMA. *Let C be a linear code with control matrix H. A set of w columns of H is linearly dependent if and only if there is a nonzero codeword in C all of whose nonzero entries occur among coordinate positions corresponding to members of that column set.*

In particular $\mathrm{d}_{\min}(C) = d$ if and only if there exists a set of d linearly dependent columns in H but no set of $d-1$ linearly dependent columns.

PROOF. All but the last sentence was discussed above. By Lemma 3.1.2 $\mathrm{d}_{\min}(C) = \mathrm{w}_{\min}(C)$. Now $\mathrm{w}_{\min}(C) \leq d$ if and only if there are d linearly dependent columns in H, while $\mathrm{w}_{\min}(C) \geq d$ if and only if all collections of $d-1$ columns are linearly independent. □

(3.1.13) PROBLEM. *Let*

$$H = \left[\mathbf{h}_1,\ \mathbf{h}_2,\ \ldots\ ,\ \mathbf{h}_n \right]$$

be the check matrix of the e-error-correcting, binary $[n,k]$ linear code D, the various $\mathbf{h}_j$ being the columns of H. Next let D' be the binary $[n,k]$ linear code with check matrix

$$H' = \left[\mathbf{h}_1,\ \mathbf{h}_1+\mathbf{h}_2,\ \mathbf{h}_1+\mathbf{h}_3,\ \ldots\ ,\ \mathbf{h}_1+\mathbf{h}_n \right] .$$

Prove that D' is also an e-error-correcting code.

(3.1.14) THEOREM. (SINGLETON BOUND.) *If C is an $[n,k]$ linear code over the field F, then*

$$\mathrm{d}_{\min}(C) \leq n-k+1 .$$

PROOF. Every $n-k \times n-k+1$ submatrix of a check matrix has rank at most $n-k$, so every set of $n-k+1$ columns of the check matrix is linearly dependent. The theorem then follows from Lemma 3.1.12. □

We have seen in Problem 2.3.10 that this result is true even for nonlinear codes. Indeed if we move k and $d = \mathrm{d}_{\min}(C)$ to opposite sides and raise $q = |F|$ to the appropriate power, we are left with

$$|C| = q^k \leq q^{n-d+1} .$$

The present proof of the bound shows that even more is true. Any set of $n-k+1$ coordinate positions contains the support (the nonzero entries) of a nonzero codeword.

(3.1.15) PROBLEM. *Use a generator matrix in* **RREF** *to give another quick proof of the Singleton bound for linear codes.*

maximum distance separable
MDS code

A linear code that meets the Singleton bound with equality is called *maximum distance separable* or, for short, an *MDS code*. Every subset of $n-k+1$ coordinate positions supports a codeword in an MDS code. By convention the zero code $\{\mathbf{0}\}$ is MDS, even though its minimum distance is somewhat ill-defined.

The $[4,2]$ ternary Hamming code has minimum distance 3 and so is MDS since $3 = 4-2+1$. We shall meet many MDS codes later when we discuss the generalized Reed-Solomon codes.

(3.1.16) PROBLEM. *Prove that the dual of an MDS codes is also an MDS code.*

(3.1.17) PROBLEM. *Prove that a binary MDS code of length n is one of $\{\mathbf{0}\}$, the repetition code, the parity check code, or all $\mathbb{F}_2^n$.*

3.2 Encoding and information

If we are transmitting with an $[n, k]$ linear code over the field F, then we think of our message as being provided as k-tuples from F, members of the space F^k. We can encode using the generator matrix G by mapping the message k-tuple $\mathbf{x}$ to the codeword $\mathbf{x}G$. Here $\mathbf{x}G$ is a codeword since, by matrix block multiplication, it is a linear combination of the rows of G (with coefficients given by $\mathbf{x}$) and $C = \mathrm{RS}(G)$.

The $k \times n$ generator matrix G is a *standard generator matrix* if its first k columns form a $k \times k$ identity matrix. The generator matrix G is *systematic* if among its columns can be found the columns of a $k \times k$ identity matrix, in which case G is said to be systematic on those columns or positions. Notice that a standard generator matrix is a special type of systematic generator matrix. If G is a standard generator, then the first k entries of the transmitted codeword $\mathbf{x}G$ contain the message vector $\mathbf{x}$. If G is systematic, then all the entries of the message vector appear among the entries of the transmitted codeword. A subset of the coordinate positions of a linear code is called an *information set* if there is a generator matrix for the code that is systematic on the columns in those positions. We can think of the positions of an information set as carrying the information, while the remaining positions are contributing redundancy. A given code may, however, have many different information sets. A choice of one set is essentially a choice of the corresponding systematic generator for encoding purposes.

standard generator matrix

systematic

information set

EXAMPLES. Consider the generator matrices given in §3.1. The generator matrix for the repetition code is (trivially) standard. For the repetition code, any coordinate position can serve as information set.

The first generator given for the parity check code is standard. Of the two further generators for the $[7, 6]$ parity check code the first is systematic but not standard, and the second is neither. Every set of 6 coordinate positions is an information set.

The generator matrix given for the $[7, 4]$ binary Hamming code is systematic. Indeed its generator matrix was designed to be systematic on the positions of the information set $\{3, 5, 6, 7\}$. Although it is not clear from our definition, the set of positions $\{1, 2, 3, 4\}$ is indeed an information set for this code, as the following standard generator matrix indicates:

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

Not every 4-subset of positions is an information set. The definition via check equations guarantees that $\{4, 5, 6, 7\}$ is not an information set, since

each codeword has an even number of 1's in these positions. Each particular even weight pattern occurs on these positions in two different codewords, while no odd weight pattern occurs at all.

The generator matrix given for the $[8, 4]$ binary extended Hamming code is systematic, but this code has no standard generator matrix since each codeword has an even number of 1's in positions $\{1, 2, 3, 4\}$.

The first generator given for the $[4, 2]$ ternary Hamming code is standard while the second is systematic but not standard. Each pair of positions is an information set. (This universality characterizes MDS codes; see Problem 3.2.3.)

It should be noted that, in some references (particularly engineering texts), generator matrices that we have called "standard" are called "systematic" and the more general class that we call "systematic" is not given a specific name.

The rows of a generator matrix form a basis of its row space, the code. Every linear code has a systematic generator matrix, for instance $\mathbf{RREF}(G)$ for any generator G, where the pivot columns are those of an identity matrix. If the code has a standard generator matrix S, then $S = \mathbf{RREF}(G)$. Therefore a code has a standard generator matrix if and only if its generator matrix G has a **RREF** in which the pivot columns are the initial columns.

(3.2.1) PROBLEM. *Let C be an $[n, k]$ linear code over F, and let J be a subset of k coordinate positions. For the generator matrix G we write G_J for the $k \times k$ matrix composed of those columns of G indexed by J. Similarly, for the codeword $\mathbf{c}$, we write $\mathbf{c}_J$ for the k-tuple of entries in the positions of J.*

The following are equivalent:

(1) *J is an information set;*

(2) *for each $\mathbf{m} \in F^k$, there is a unique $\mathbf{c} \in C$ with $\mathbf{c}_J = \mathbf{m}$;*

(3) *for every generator matrix G, the matrix G_J is invertible.*

(3.2.2) PROBLEM. *For a nonlinear code C over A, define an information set to be a minimal subset J of the coordinate positions such that no member of $A^{|J|}$ is repeated in these positions. Prove that the dimension of C is a lower bound for $|J|$.*

Two related codes may be different but still share many properties. For instance, if the code D is gotten from C by reversing all codewords (*i.e.*, first entry last, ... , last entry first) then C and D will likely be different but will have many common properties—the same length, minimum distance, dimension, etc. For many purposes we need not distinguish between C and D.

permutation equivalent

Two codes C and D of length n over A are *permutation equivalent* if they are the same up to a uniform permutation of their coordinate entries. (This is often abbreviated to *equivalence*.) That is, there is a permutation σ of the set $\{1, \ \dots \ , \ n\}$ such that

equivalence

$$(x_1, x_2, \ \dots \ , \ x_n) \in C \iff (x_{\sigma(1)}, x_{\sigma(2)}, \ \dots \ , \ x_{\sigma(n)}) \in D.$$

Since every linear code has a systematic generator matrix, and a systematic matrix can be changed into a standard matrix by permuting columns, we find that every linear code is equivalent to a code with a standard generator matrix.

Although we do not need the definitions right now, this seems a good time to give three further notions of equivalence for codes defined over fields. Notice that linearity is not required for the various forms of equivalence. In practice regular equivalence is the one of most relevance for codes that are not linear.

DEFINITION. Two codes C and D of length n over the field F are *diagonally equivalent* if they are the same up to the multiplication in each codeword of the i^{th} entry by the nonzero constant α_i, for each i. diagonally equivalent

DEFINITION. Two codes C and D of length n over the field F are *monomially equivalent* if they are the same up to: monomially equivalent
(1) a uniform permutation of their entries (as with regular equivalence);
(2) the multiplication in each codeword of the i^{th} entry by the nonzero constant α_i, for each i.

So monomial equivalence is the union of regular equivalence and diagonal equivalence. For a linear code it corresponds to multiplying column i of a generator matrix by the constant α_i in addition to permuting the columns of the generator.

DEFINITION. Two codes C and D of length n over the field F are *affine equivalent* if they are the same up to: affine equivalent
(1) a uniform permutation of their entries;
(2) the multiplication in each codeword of the i^{th} entry by the nonzero constant α_i, for each i;
(3) translation by a fixed vector of F^n.

Two codes that are affine equivalent have the same size, length, and minimum distance. Here if C is a linear code, then D is a coset of a code monomially equivalent to C.

(3.2.3) PROBLEM. *For $k \neq 0$, prove that the $[n,k]$ linear code C is an MDS code if and only if every subset of k coordinate positions is an information set.*

(3.2.4) PROBLEM. (THRESHOLD DECODING OF MDS CODES.) *Let C be an $[n,k]$ linear MDS code with $k \neq 0$ and generator matrix G. For a set J of coordinate positions, the matrix G_J is that submatrix of G composed of the columns of G that are indexed by the members of J.*

By Problem 3.2.3 every k subset J of coordinate positions is an information set for C, so by Problem 3.2.1 the matrix G_J is always invertible. Indeed if the message k-tuple $\mathbf{m}$ gives rise to the codeword $\mathbf{c} = \mathbf{m}G$, then we can recover $\mathbf{m}$ from $\mathbf{c}$ by $\mathbf{m} = \mathbf{c}_J G_J^{-1}$.

For decoding purposes this means that, for any received vector $\mathbf{r}$, each k subset J of coordinate positions produces a "guess" or "vote" $\widehat{\mathbf{m}}_J = \mathbf{r}_J G_J^{-1}$ as to the identity of the original message $\mathbf{m}$. We choose that k-tuple $\widehat{\mathbf{m}}$ that receives the most votes and then decode $\mathbf{r}$ to the codeword $\widehat{\mathbf{c}} = \widehat{\mathbf{m}}G$.

Suppose that $\mathbf{c} = \mathbf{m}G$ has been transmitted and that $\mathbf{r}$ has been received, e symbol errors having occurred (that is, $\mathrm{d_H}(\mathbf{c}, \mathbf{r}) = e$). For k independent variables $x_1, \ldots, x_k$ arranged as a row vector $\mathbf{x} = (x_1, \ldots, x_k)$, consider the n linear equations, $j = 1, \ldots, n$,

$$\mathsf{Eqn}_j : \qquad \mathbf{r}_j = \mathbf{x}G_j \ ,$$

where $\mathbf{r}_j$ *is the* j*-th entry of received* $\mathbf{r}$ *and* G_j *is the* j*-th column of the matrix* G*.*

(*a*) *Prove that setting* $\mathbf{x}$ *equal to* $\mathbf{m}$ *solves* $n-e$ *of the equations* Eqn_j*. Prove that* $\mathbf{m}$ *gets* $\binom{n-e}{k}$ *votes.*

(*b*) *For any* k*-tuple* $\mathbf{l}$ *that is not equal to* $\mathbf{m}$*, prove that setting* $\mathbf{x}$ *equal to* $\mathbf{l}$ *solves at most* $e+k-1$ *of the equations* Eqn_j*. Prove that* $\mathbf{l}$ *gets at most* $\binom{e+k-1}{k}$ *votes.*

(*c*) *Prove that, as long as* $2e < n-k+1$ $(= \mathrm{d}_{\min}(C))$*, the received vector* $\mathbf{r}$ *will be decoded correctly to* $\mathbf{c}$*.*

(3.2.5) PROBLEM. *Consider the MDS code* C *over the field* $\mathbb{F}_7$ *of integers modulo* 7 *with generator matrix*

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 4 & 6 & 1 & 3 & 5 \end{bmatrix}.$$

Use the method of the previous problem to decode the received vector

$$\mathbf{r} = (1,3,6,5,4,2)\,.$$

3.3 Decoding linear codes

dictionary decoding

A form of decoding always available is *dictionary decoding*. In this we make a list of all possible received words and next to each word write the codeword (or codewords) to which it may be decoded under **MLD**. In decoding, when a word is received we look it up in our "dictionary" and decode it to a codeword listed opposite it. This will almost never be a practical solution.

We now wish to use the structure of linear codes to aid in their decoding. If we are transmitting with a linear code C of length n over F, then we can think of the channel as adding in an *error vector* or *error word* (or even *error pattern* in the binary case). If $\mathbf{c}$ is transmitted and $\mathbf{x}$ is received, then the channel noise has had the effect of adding to $\mathbf{c}$ the error vector $\mathbf{e} = \mathbf{x} - \mathbf{c} \in F^n$, so that $\mathbf{x} = \mathbf{c} + \mathbf{e}$. The decoding problem is then, given $\mathbf{x}$, estimate at least one of $\mathbf{c}$ and $\mathbf{e}$. The weight of $\mathbf{e}$ is the number of positions in which $\mathbf{c}$ and $\mathbf{x}$ differ; so, when using an $m\mathrm{SC}(p)$ (with $p < 1/m$) and decoding under **MLD**, we are looking for an error pattern $\mathbf{e}$ of minimum weight.

error vector
error word
error pattern

From the definition of $\mathbf{e}$, we learn that the received vector $\mathbf{x}$ and the error pattern $\mathbf{e}$ belong to the same coset $\mathbf{x} + C = \mathbf{e} + C$. While we do not know $\mathbf{x}$ ahead of time, the cosets of C can be calculated in advance. We look for vectors of minimum weight in each coset. Such a vector is caller a *coset leader*. Notice that while the minimum weight of a coset is well-defined, there may be more than one vector of that weight in the coset, that is, there may be more than one coset leader. Usually we choose and fix one of the coset leaders. Always $\mathbf{0}$ is the unique coset leader for the code itself.

coset leader

We first describe the general technique of decoding with coset leaders and then give two methods for its implementation. When the word $\mathbf{x}$ is received, we do not know the actual error that was introduced; but we do know that it belongs to the coset $\mathbf{x} + C$. Thus if $\widehat{\mathbf{e}}$ is the coset leader chosen for this coset, then $\widehat{\mathbf{e}}$ is one of the most likely error patterns; and we guess that it was the actual error. (In fact $\widehat{\mathbf{e}}$ is the unique most likely error pattern if it is the unique

leader of its coset.) We decode $\mathbf{x}$ to the codeword $\widehat{\mathbf{c}} = \mathbf{x} - \widehat{\mathbf{e}}$. With coset leader decoding, the error patterns that are corrected are exactly those that are the chosen coset leaders. In particular, the code will be e-error-correcting if and only if every vector of weight at most e is the unique leader of its coset.

Coset leader decoding is an **MDD** algorithm for linear codes over fields of size m. Therefore knowledge of the coset leaders for C makes it easy to calculate $\mathcal{P}_C$ on an $m\mathrm{SC}(p)$. Indeed, an error pattern will be corrected if and only if it is a chosen coset leader. Thus, for $m\mathrm{SC}(p)$ with $q = 1 - (m-1)p > p$, we have

$$\mathcal{P}_C = \mathcal{P}_C(\mathbf{MDD}) = 1 - \left(\sum_{i=0}^{n} a_i p^i q^{n-i} \right),$$

where a_i is the number of cosets of C with coset leader of weight i.

(3.3.1) PROBLEM. *If* $\mathbf{x}$ *and* $\mathbf{y}$ *are binary vectors of length* n*, then we write* $\mathbf{x} \preceq \mathbf{y}$ *to indicate that* $\mathbf{x}$ *has a* 0 *in every position that* $\mathbf{y}$ *has a* 0 *(but* $\mathbf{x}$ *may have* 0*'s in places that* $\mathbf{y}$ *has* 1*'s.) Equivalently, everywhere* $\mathbf{x}$ *has a* 1*,* $\mathbf{y}$ *also has a* 1*, but* $\mathbf{y}$ *may have more* 1*'s. For instance*

$$(0,0,0,1,1,0,1,0) \preceq (0,1,0,1,1,0,1,1)\,.$$

(*a*) *Let* $\mathbf{x}$ *and* $\mathbf{y}$ *be binary* n*-tuples, and set* $\mathbf{f} = \mathbf{x} + \mathbf{y}$*. Prove that* $\mathbf{x} \preceq \mathbf{y}$ *if and only if* $\mathrm{w_H}(\mathbf{y}) = \mathrm{w_H}(\mathbf{x}) + \mathrm{w_H}(\mathbf{f})$*.*

(*b*) *Let* C *be a binary linear code of length* n*. Prove that if* $\mathbf{y}$ *is a coset leader for the coset* $\mathbf{y} + C$ *and* $\mathbf{x} \preceq \mathbf{y}$*, then* $\mathbf{x}$ *is also a coset leader for the coset* $\mathbf{x} + C$*.*

standard array decoding

Our first method for coset leader decoding is *standard array decoding*. Set $K = |C|$, the cardinality of C, and $R = |F^n|/|C|$, the number of distinct cosets of C. Enumerate the codewords:

$$C = \{\ \mathbf{c}_1 = \mathbf{0},\ \mathbf{c}_2,\ \ldots\ , \mathbf{c}_K\}$$

and coset leaders:

$$\{\ \mathbf{e}_1 = \mathbf{0},\ \mathbf{e}_2,\ \ldots\ , \mathbf{e}_R\},$$

one coset leader for each coset of C in F^n. We form a large array, the *standard array*, whose first row contains the codewords, and first column contains the coset leaders, and in general has $\mathbf{c}_j + \mathbf{e}_i$ as its i, j entry. The i^{th} row of the standard array is the coset $\mathbf{e}_i + C$. Thus every n-tuple of F^n is contained exactly once in the array.

To decode using the standard array, when $\mathbf{x}$ is received, look it up in the array. If it is in the i, j position, then we have $\mathbf{x} = \mathbf{c}_j + \mathbf{e}_i$. In this case we assume that the introduced error vector was $\mathbf{e}_i$ and decode $\mathbf{x}$ to $\widehat{\mathbf{c}} = \mathbf{c}_j$.

Standard array decoding is not of much practical value as it involves storage of the large array as well as random access searches through the array. It does have historical and theoretical value, because it illustrates the important general fact that code structure can be exploited to design decoding algorithms that are more efficient than dictionary decoding.

syndrome decoding

The second method of coset leader decoding is *syndrome decoding*, where the dual code and check matrices are used. Let H be a check matrix for the $[n, k]$ linear code C. We have already mentioned that the vector $\mathbf{x}$ is in the code C if and only if the matrix product $H\mathbf{x}^\top$ equals $\mathbf{0}$. For any received vector $\mathbf{x}$, the length $r = n - k$ column vector $H\mathbf{x}^\top$ is a measure of whether on not the n-tuple $\mathbf{x}$ belongs to the code. The column r-tuple $H\mathbf{x}^\top$ is the *syndrome* of the n-tuple $\mathbf{x}$. According to the "Pocket Oxford Dictionary," a syndrome is generally a "characteristic combination of opinions." The syndrome voices information about the error vector. Syndrome decoding is error oriented, using the opinions voiced by the syndrome vector to identify the appropriate error vector.

syndrome

As the syndrome of a codeword is $\mathbf{0}$, two vectors $\mathbf{x}$ and $\mathbf{e}$ that differ by a codeword $\mathbf{c}$ will have the same syndrome:

$$H\mathbf{x}^\top = H(\mathbf{c} + \mathbf{e})^\top = \mathbf{0} + H\mathbf{e}^\top = H\mathbf{e}^\top$$

That is, syndromes are constant on cosets of C in F^n. Equally well, distinct cosets have different syndromes since the difference of vectors from distinct cosets is not a codeword and so has nonzero syndrome.

We interpret the above equation as saying that a received vector $\mathbf{x}$ and the corresponding error vector $\mathbf{e}$ introduced by the channel will have the same syndrome, namely that of the coset to which they both belong. Instead of storing the entire standard array, we need only store a *syndrome dictionary* (or syndrome table) containing all possible syndromes $\{\mathbf{s}_1 = \mathbf{0}, \ \ldots \ , \ \mathbf{s}_R\}$ together with coset leaders $\{\mathbf{e}_1 = \mathbf{0}, \ \ldots \ , \ \mathbf{e}_R\}$ such that $H\mathbf{e}_i^\top = \mathbf{s}_i$. In decoding, when $\mathbf{x}$ is received, first calculate the syndrome $\mathbf{s} = H\mathbf{x}^\top$. Next look up $\mathbf{s}$ in the syndrome dictionary as $\mathbf{s} = \mathbf{s}_i$. Finally decode $\mathbf{x}$ to $\widehat{\mathbf{c}} = \mathbf{x} - \mathbf{e}_i$.

syndrome dictionary

EXAMPLE. Consider the $[4, 2]$ ternary Hamming code with check matrix

$$H = \begin{bmatrix} 1 & 1 & 2 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} .$$

The syndromes are therefore column vectors of length 2. For instance, the received vector $\mathbf{x} = (1, 2, 1, 1)$ has syndrome

$$H\mathbf{x}^\top = \begin{pmatrix} 1+2+2+0 \\ 0+2+1+1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} .$$

To decode using syndromes we first write out our syndrome dictionary,

the first column containing the transposes of all possible syndromes.

syndrome transpose	coset leader
00	0000
01	0001
02	0002
10	1000
11	0100
12	0020
20	2000
21	0010
22	0200

It is not necessary to list out all cosets of the code to make this dictionary. Instead notice that two words of $\mathbb{F}_3^4$ are in the same coset if and only if their difference is a codeword. So, for instance, not only must $(0,0,0,1)$, $(0,0,0,2)$, and $(0,2,0,0)$ all be of minimum weight in their respective cosets; but they belong to different cosets. (Subtracting one of them from another gives a word of weight less than 3, not a codeword since the minimum weight of the Hamming code is 3.) The transposed syndromes are then calculated as, respectively, $(0,1)$, $(0,2)$, and $(2,2)$; and the results are recorded in the dictionary.

To decode our received vector $\mathbf{x} = (1,2,1,1)$ we first calculate, as before, its transposed syndrome $(2,1)$. We then look up this syndrome in our dictionary and discover that the corresponding coset leader is $\widehat{\mathbf{e}} = (0,0,1,0)$. We therefore assume that this is the error that occurred and decode $\mathbf{x}$ to the codeword

$$\widehat{\mathbf{c}} = \mathbf{x} - \widehat{\mathbf{e}} = (1,2,1,1) - (0,0,1,0) = (1,2,0,1) \ .$$

It may sometimes be more convenient to define syndromes and do syndrome decoding relative to a control matrix H rather than a check matrix.

Syndrome decoding does not suffer from many of the failings of standard array decoding. The syndrome dictionary is much smaller than the standard array for storage purposes; and it can be ordered lexicographically, so that searches can be done linearly. Still syndrome decoding in this dictionary form is too general to be of much practical use. Certain practical decoding algorithms do employ partial syndrome dictionaries that list only the most common syndromes. Syndrome decoding is also the paradigm for many genuine decoding techniques. To each received vector we associate some kind of "syndrome." The properties of the specific code then are used in the passage from syndrome to error vector and decoded word. The decoding method for the $[7,4]$ Hamming code as given by Shannon in Example 1.3.3 is a type of syndrome decoding, since he has arranged the columns of the check matrix H (given on page 37) to contain the binary numbers in order. The calculated syndrome $\alpha\beta\gamma$ is therefore associated with the coset whose leader has a 1 in the $\alpha\beta\gamma^{\text{th}}$ position (read in binary) and 0 elsewhere. We decode assuming an error in this position.

(3.3.2) PROBLEM. (*a*) *Give a syndrome dictionary for the* $[8,4]$ *extended binary Hamming code* E *with the following check matrix (and generator matrix—the code is self dual):*

$$\mathrm{XL}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} .$$

(*b*) *Use your table to decode the received word:*

$$(0,0,1,0,0,1,1,0) .$$

(*c*) *Use your table to decode the received word:*

$$(0,1,1,1,1,1,0,1) .$$

We now consider using syndrome decoding and check matrices to correct erasures rather than errors. (See Problem 2.2.4.) Remember that erasures occur as a consequence of soft quantization of received symbols. We allow transmitted symbols from the alphabet A to be received as members of A or as **?**, the erasure symbol. Alternatively we may think of erasures as symbol errors whose locations are known. Under this second interpretation, we might receive a word from the alphabet A but with certain positions flagged as being unreliable. These flagged positions are then the erasure locations. The two views of erasures are equivalent. Indeed each occurrence of **?** may filled arbitrarily by an alphabet letter (typically 0 for a linear code) and then flagged as unreliable. Conversely each flagged symbol can be replaced by **?**, the erasure symbol. Which point of view is the best will depend upon the particular situation.

Since C contains codewords of weight $d = \mathrm{d}_{\min}(C)$ as well as $\mathbf{0}$ of weight 0, we could never hope to correct d erasures; but we can decode up to $d-1$ erasures correctly.

(3.3.3) PROPOSITION. *Let* C *be an* $[n,k,d]$ *linear code over* F *with check matrix* H *whose rows are* $\mathbf{h}_i$*, for* $i = 1,\ldots,r = n-k$*. Let* $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ *be an* n*-tuple of indeterminates.*

Assume the codeword $\mathbf{c} = (c_1, c_2, \ldots, c_n)$ *is transmitted, and we receive the vector* $\mathbf{p} = (p_1, p_2, \ldots, p_n) \in F^n$ *with the entries* p_l*, for* $l \in L$*, flagged as erasures but* $p_j = c_j$*, for* $j \notin L$*.*

If $|L| \leq d-1$*, then the set of equations in the unknowns* x_i

$$\begin{aligned} \mathbf{h}_i \cdot \mathbf{x} &= \mathbf{h}_i \cdot \mathbf{p} \text{ for } i = 1,\ldots,r \\ x_j &= 0 \text{ for } j \notin L \end{aligned} \qquad (*)$$

has as its unique solution the erasure vector

$$\mathbf{x} = \mathbf{c} - \mathbf{p} = \mathbf{e} .$$

Therefore by solving the equations $(*)$ *we can decode all patterns of up to* $d-1$ *erasures in codewords of* C*.*

PROOF. This set of equations has at least one solution, namely the actual erasure vector $\mathbf{e} = \mathbf{c} - \mathbf{p}$. If $\mathbf{e}'$ is any solution of the equations $(*)$ then $\mathbf{c}' = \mathbf{e} - \mathbf{e}'$ has syndrome $\mathbf{0}$ and equals 0 off L. Therefore $\mathbf{c}'$ is a codeword of weight at most $d - 1$ and so must be $\mathbf{0}$. We conclude that $\mathbf{e} = \mathbf{e}'$, and the set of equations $(*)$ has the unique solution $\mathbf{x} = \mathbf{e}$. □

The equations $(*)$ give $n + r - |L|$ linear equations in the n unknowns, where $r \geq d - 1 \geq |L|$ (by the Singleton bound 3.1.14). By the proposition, the solution is unique; so the system has rank n. The last $n - |L|$ syndrome equations of $(*)$ are clearly linearly independent; so we may delete some of the first r equations to reduce $(*)$ to a system of n equations in n unknowns with a unique solution. These equations can be solved by Gaussian elimination; so the number of operations required is at worst on the order of n^3, a respectably small number. Indeed the set of equations is essentially triangular, so the complexity is actually on the order of n^2.

The algorithm of the proposition is an effective method for correcting erasures in any linear code. This can in turn be helpful when decoding errors as well as erasures. We may concentrate our efforts upon designing an algorithm that locates errors. After the errors have been located, they can be thought of as flagged erasures and their values found with the algorithm of Proposition 3.3.3.