**Generator Matrices**

We can arrange a set of basis vectors for a linear code in a *generator matrix*, each row of which is a basis vector.

A generator matrix for an \([n, k]\) code will have \(k\) rows and \(n\) columns.

Here's a generator matrix for the \([5, 2]\) code looked at earlier:

\[
\begin{pmatrix}
0 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 1
\end{pmatrix}
\]

Note: Almost all codes have more than one generator matrix.

**Encoding Blocks Using a Generator Matrix**

We can use a generator matrix for an \([n, k]\) code to encode a block of \(k\) message bits as a block of \(n\) bits to send through the channel.

We regard the \(k\) message bits as a row vector, \(\mathbf{a}\), and multiply by the generator matrix, \(\mathbf{G}\), to produce the channel input, \(\mathbf{u}\):

\[
\mathbf{u} = \mathbf{aG}
\]

If the rows of \(\mathbf{G}\) are linearly independent, each \(\mathbf{a}\) will produce a different \(\mathbf{u}\), and every \(\mathbf{u}\) that is a codeword will be produced by some \(\mathbf{a}\).

**Example:** Encoding the message block \((1, 1)\) using the generator matrix for the \([5, 2]\) code given earlier:

\[
\begin{pmatrix}
1 \\
1
\end{pmatrix}
\begin{pmatrix}
0 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 1
\end{pmatrix} =
\begin{pmatrix}
1 \\
1 \\
1 \\
1 \\
1
\end{pmatrix}
\]

**Parity-Check Matrices**

Suppose we have specified an \([n, k]\) code by a set of \(c = n - k\) equations satisfied by any codeword, \(\mathbf{v}\):

\[
\begin{align*}
\sum b_{1,1} v_1 + b_{1,2} v_2 + \cdots + b_{1,n} v_n &= 0 \\
\sum b_{2,1} v_1 + b_{2,2} v_2 + \cdots + b_{2,n} v_n &= 0 \\
\vdots \\
\sum b_{c,1} v_1 + b_{c,2} v_2 + \cdots + b_{c,n} v_n &= 0
\end{align*}
\]

We can arrange the coefficients in these equations in a *parity-check matrix*, as follows:

\[
\begin{pmatrix}
b_{1,1} & b_{1,2} & \cdots & b_{1,n} \\
b_{2,1} & b_{2,2} & \cdots & b_{2,n} \\
\vdots \\
b_{c,1} & b_{c,2} & \cdots & b_{c,n}
\end{pmatrix}
\]

If \(\mathcal{C}\) has parity-check matrix \(\mathbf{H}\), we can check whether \(\mathbf{v}\) is in \(\mathcal{C}\) by seeing whether \(\mathbf{vH}^T = \mathbf{0}\).

Note: Almost all codes have more than one parity-check matrix.

**Example: The \([5, 2]\) Code**

Here is one parity-check matrix for the \([5, 2]\) code used earlier:

\[
\begin{pmatrix}
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 & 1
\end{pmatrix}
\]

We see that 11001 is a codeword as follows:

\[
\begin{pmatrix}
1 \\
1 \\
0
\end{pmatrix}
\begin{pmatrix}
1 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 1
\end{pmatrix} =
\begin{pmatrix}
0 & 0 & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

But 10011 isn't a codeword, since

\[
\begin{pmatrix}
1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
1 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 1 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix} =
\begin{pmatrix}
1 & 1 & 0
\end{pmatrix}
\]
Examples: Repetition Codes and Single Parity-Check Codes

An \([n, 1]\) repetition code has the following generator matrix (for \(n = 4\)):

\[
\begin{pmatrix}
1 & 1 & 1 & 1
\end{pmatrix}
\]

Here is a parity-check matrix for this code:

\[
\begin{pmatrix}
1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1
\end{pmatrix}
\]

One generator matrix for an \([n, n-1]\) single parity-check code is the following:

\[
\begin{pmatrix}
1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1
\end{pmatrix}
\]

Here is the parity-check matrix for this code:

\[
\begin{pmatrix}
1 & 1 & 1 & 1
\end{pmatrix}
\]

Dual Codes

If \(C\) is a linear \([n, k]\) code, the set of all vectors orthogonal to every vector in \(C\) is a linear \([n, n - k]\) code — the dual of \(C\) (written \(C^\perp\)).

Why is \(C^\perp\) a linear code? If \(v_1\) and \(v_2\) are in \(C^\perp\), then \(v_1 \cdot u = 0\) and \(v_2 \cdot u = 0\) for every \(u\) in \(C\). Hence \((v_1 + v_2) \cdot u = 0\) for every \(u\) in \(C\), from which it follows that \(v_1 + v_2\) is in \(C^\perp\).

Suppose \(u_1, \ldots, u_k\) is a set of basis vectors for \(C\). A vector \(v\) will be orthogonal to all \(u\) in \(C\) if and only if it is orthogonal to all these basis vectors. In other words:

\[
v \cdot u_1 = 0 \& v \cdot u_2 = 0 \& \cdots \& v \cdot u_k = 0
\]

if and only if

\[
v \cdot (a_1 u_1 + a_2 u_2 + \cdots + a_k u_k) = 0 \quad \text{for all } a_1, \ldots, a_k
\]

It follows that \(C^\perp\) is an \([n, n - k]\) code, since its codewords satisfy \(k\) independent equations.

Generator and Parity-Check Matrices For Dual Codes

Suppose \(C\) has a generator matrix \(G\) and a parity-check matrix \(H\).

A vector \(v\) will be in \(C^\perp\) if and only if it is orthogonal to all the rows of \(G\) — in other words, if \(vG^T = 0\). So \(G\) is a parity-check matrix for \(C^\perp\).

If \(v\) is a row of \(H\), it must be in \(C^\perp\), since \(v \cdot u = 0\) for every \(u\) in \(C\). The rows of \(H\) are independent, so these \(n - k\) rows form a basis for \(C^\perp\). Hence \(H\) is a generator matrix for \(C^\perp\).

We can get the dual of a code by swapping its generator and parity-check matrices. The repetition and single-parity check codes are each duals of the other.

In general, the dual of the dual of \(C\) is \(C\) itself. Some codes are their own duals.

Manipulating the Parity-Check Matrix

There are usually many parity-check matrices for a given code. We can get one such matrix from another using the following “elementary row operations”:

- Swapping two rows.
- Multiplying a row by a non-zero constant (not useful for \(F_2\)).
- Adding a row to a different row.

These operations don’t alter the solutions to the equations the parity-check matrix represents.

Ex: This parity-check matrix for the \([5, 2]\) code:

\[
\begin{pmatrix}
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 & 1
\end{pmatrix}
\]

can be transformed into this alternative:

\[
\begin{pmatrix}
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 1
\end{pmatrix}
\]
**Manipulating the Generator Matrix**

We can apply the same elementary row operations to a generator matrix for a code, in order to produce another generator matrix, since these operations just convert one set of basis vectors to another.

**Example:** Here is a generator matrix for the [5,2] code we have been looking at:

\[
\begin{pmatrix}
0 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 1
\end{pmatrix}
\]

Here is another generator matrix, found by adding the first row to the second:

\[
\begin{pmatrix}
0 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0
\end{pmatrix}
\]

**Note:** These manipulations leave the set of codewords unchanged, but they don’t leave the way we encode messages by computing \( \mathbf{u} = \mathbf{aG} \) unchanged!

**Equivalent Codes**

Two codes are said to be equivalent if the codewords of one are just the codewords of the other with the order of symbols permuted.

Permuting the order of the columns of a generator matrix will produce a generator matrix for an equivalent code, and similarly for a parity-check matrix.

**Example:** Here is a generator matrix for the [5,2] code we have been looking at:

\[
\begin{pmatrix}
0 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 1
\end{pmatrix}
\]

We can get an equivalent code using the following generator matrix obtained by moving the last column to the middle:

\[
\begin{pmatrix}
0 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0
\end{pmatrix}
\]

---

**Generator and Parity-Check Matrices In Systematic Form**

Using elementary row operations and column permutations, we can convert any generator matrix to a generator matrix for an equivalent code that is is systematic form, in which the left end of the matrix is the identity matrix.

Similarly, we can convert to the systematic form for a parity-check matrix, which has an identity matrix in the right end.

For the [5,2] code, only permutations are needed. The generator matrix can be permuted by swapping columns 1 and 3:

\[
\begin{pmatrix}
0 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 1
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
1 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 1
\end{pmatrix}
\]

When we use a systematic generator matrix to encode a block \( \mathbf{a} \) as \( \mathbf{u} = \mathbf{aG} \), the first \( k \) bits will be the same as those in \( \mathbf{a} \). The remaining \( n - k \) bits can be seen as “check bits”.

---

**Relationship of Generator and Parity-Check Matrices**

If \( G \) and \( H \) are generator and parity-check matrices for \( \mathcal{C} \), then for every \( \mathbf{a} \), we must have \( (\mathbf{aG})H^T = \mathbf{0} \) — since we should only generate valid codewords. It follows that

\[
GH^T = \mathbf{0}
\]

Furthermore, any \( H \) with \( n - k \) independent rows that satisfies this is a valid parity-check matrix for \( \mathcal{C} \).

Suppose \( G \) is in systematic form, so

\[
G = [I_k \mid P]
\]

for some \( P \). Then we can find a parity-check matrix for \( \mathcal{C} \) in systematic form as follows:

\[
H = [-P^T \mid I_{n-k}]
\]

since \( GH^T = -I_k P + P I_{n-k} = \mathbf{0} \). (Note that \(-P^T = P^T \) in \( F_2 \).)