MATH 525
Section 2.6: Generating Matrices and Encoding

September 28, 2020

Goal: Define a generating (or generator) matrix of a linear code and show how it is used for encoding messages. The process is faster and "much simpler" than that for arbitrary nonlinear codes.

Definition

Let C be a linear code of length n. Then:

- Any matrix G whose rows form a basis for C is called a generating matrix for C.
- The number of rows of G is called the rank of G. This number, denoted by k, is the dimension of C.

Terminology: If C is a linear code of length n, dimension k, and distance d, we refer to it as an (n, k, d) linear code. These three parameters give a good measure of how good C is. In this case, $G = (g_{ij})_{k \times n}$.

Remark: The dimension of C is the dimension of C as a subspace of K^n.

Remark: A linear code C usually has many different generating matrices for if G is a generating matrix, then any matrix that is row equivalent to G is also a generating matrix for C. However, there is exactly one generating matrix in RREF.

Encoding of Linear Codes:

Let C be a linear code with generating matrix G of size $k \times n$. The long message (string of 0s and 1s) which comes out of the source is broken down into blocks of k symbols. Each block $\mathbf{u} = (u_1, \ldots, u_k) \in K^k$ is encoded as:

$$\mathbf{u} \mapsto \mathbf{u}G.$$

The codeword $\mathbf{v} = \mathbf{u}G$ is sent through the channel. We call $\mathbf{u}$ the information vector and $\mathbf{v} = \mathbf{u}G$ the codeword corresponding to $\mathbf{u}$.

There are 2^k codewords in C and each corresponds to a unique information vector in K^k. In symbols: $\mathbf{u}_1 G = \mathbf{u}_2 G$ if and only if $\mathbf{u}_1 = \mathbf{u}_2$.

We can already see that it is much easier to implement the encoder of a linear code than the encoder of a nonlinear code: The encoder of a linear code of dimension k requires the storage of only k of the 2^k codewords. This represents tremendous savings!

Example

Consider a code C over K with generator matrix

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

The message $\mathbf{u} = (u_1, u_2, u_3)$ is encoded as

$$(u_1, u_2, u_3) \cdot \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} = (u_1, u_2 + u_3, u_1 + u_2, u_3, u_1 + u_3).$$

N.B.: All operations are modulo 2, that is, they occur in the field $K = \{0, 1\}$.

Equivalent Codes and Systematic Encoding

Let $G = (g_{ij})_{k\times n}$, $k < n$, be such that

$$G = [I_k | X].$$

G is said to be in standard or systematic form and the code generated by G is a systematic code.

Not all codes have a generating matrix in systematic form, e.g., the code whose generating matrix is:

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

(why?).

Why is the systematic form interesting?

Suppose $G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$ and we wish to encode the information vector $\mathbf{u} = (u_1, u_2, u_3, u_4)$. We get

$$\begin{aligned} \mathbf{v} = \mathbf{u}G &= (u_1, u_2, u_3, u_4)G = \\ &= (u_1, u_2, u_3, u_4, u_1 + u_3 + u_4, u_1 + u_2 + u_3, u_2 + u_3 + u_4). \end{aligned}$$

It is not difficult to see that in general, if $G = [I_k|X]$ and $\mathbf{u} = (u_1, \ldots, u_k)$, then

$$\begin{aligned} \mathbf{v} = \mathbf{u}G &= (v_1, \ldots, v_k, v_{k+1}, \ldots, v_n) = \\ &= (u_1, \ldots, u_k, v_{k+1}, \ldots, v_n). \end{aligned}$$

Conclusion: In systematic encoding, the first block of k bits of every codeword is the corresponding information vector. The remaining $n-k$ bits are called the redundant bits or parity-check bits.

In summary, if the generating matrix is in systematic form:

1. Encoding is generally less complex (from the hardware or software point of view);
2. When the decoder decides that a certain received word $\mathbf{r} = (r_1, r_2, \ldots, r_n)$ is a codeword, then it can quickly obtain the corresponding information vector just by extracting the first k bits from $\mathbf{r}$.

Otherwise, if a non-systematic code is used, then once the decoder decides that a certain received word $\mathbf{r} = (r_1, r_2, \ldots, r_n)$ is a codeword, then it needs to solve the system $\mathbf{r} = \mathbf{u}G$ in order to determine $\mathbf{u}$. Recall that the user at the receiving end only cares about information vectors (and not codewords).

Definition

Two codes C_1 and C_2 are said to be equivalent if C_2 can be obtained from C_1 through a (fixed) permutation of the coordinates in each codeword of C_1.

Example

$C_1 = \{0000, 0011, 1100, 1111\}$ and $C_2 = \{0000, 0101, 1010, 1111\}$ are equivalent codes. The permutation

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & 2 & 4 \end{pmatrix}$$

(applied on each codeword) can be used to transform C_1 into C_2.

Equivalent codes have the same length, dimension, and minimum distance. Their performances are identical.

Theorem

Any linear code C is equivalent to a linear code C' having a generating matrix in standard (or systematic) form.

Outline of Proof: Let G be a generating matrix for C. Place G in RREF (if it is not already). Permute the columns of the obtained matrix so that the leading columns come first and form an identity matrix. $\square$