Week 3

Linear codes

Version 2023-10-02. To accessible online version of this chapter

Synopsis. *The most important class of codes is linear codes. Their ability to correct errors is no worse than that of general codes, but linear codes are easier to implement in practice and allow us to use algebraic methods. We learn how to find the minimum distance by looking at weights, and how to define a linear code by its generator matrix.*

The definition of a linear code

Reminder (vector spaces): let $\mathbb{F}_q$ denote the field of q elements. When we use $\mathbb{F}_q$ as the alphabet, we refer to words in $\mathbb{F}_q^n$ as (row) **vectors**. The set $\mathbb{F}_q^n$ of all vectors of length n has the structure of a **vector space** over the field $\mathbb{F}_q$. If the vectors $\underline{u}$, $\underline{v}$ are in $\mathbb{F}_q^n$, we can add the vectors together: $\underline{u} + \underline{v} \in \mathbb{F}_q^n$, and multiply a vector by a scalar: $\lambda\underline{u} \in \mathbb{F}_q^n$ for all $\lambda \in \mathbb{F}_q$. The addition and the scalar multiplication are performed **componentwise**. We will often write vectors in compact form, as words:

$$011011, 100110 \in \mathbb{F}_2^6 \qquad \mapsto \qquad 011011 + 100110 = 111101 \in \mathbb{F}_2^6.$$

Definition: linear code, codevector

A **linear code** is a subspace of the vector space $\mathbb{F}_q^n$.
Codewords of a linear code are called **codevectors.**

This means that the zero vector $\underline{0}$ belongs to C, and that sums and scalar multiples of codevectors are again codevectors. Thus, C is a vector space in its own right.

Discussion: Why are linear codes useful? (not examinable)

1. They seem to be as efficient as general codes. In particular, it was proved that Shannon's Theorem about the capacity of a channel (discussed later) is still true for linear codes.

2. It is possible to define a linear code without specifying all the codewords (see below).

3. The minimum distance is easier to calculate than for general codes (see below).

4. We can use algebra to design linear codes and to construct efficient encoding and decoding algorithms.

The absolute majority of codes designed by coding theorists are linear codes. In the rest of the course, (almost) all the codes we consider will be linear codes.

End of discussion.

Example: trivial, repetition codes

The trivial code $\mathbb{F}_q^n$ is a linear code. (Indeed, $\mathbb{F}_q^n$ is a vector subspace of itself.)
The repetition code $Rep(n, \mathbb{F}_q)$ over $\mathbb{F}_q$ is a linear code (*exercise; will see soon*).

To get non-trivial examples, we need to introduce more structure.

The weight

Definition: weight of a vector, weight of a code

The **weight** $w(\underline{v})$ **of a vector** $\underline{v} \in \mathbb{F}_q^n$ is the number of non-zero symbols in $\underline{v}$.
The **weight** $w(C)$ **of a code** $C \subseteq \mathbb{F}_q^n$ is $w(C) = \min\{w(\underline{v}) \mid \underline{v} \in C \setminus \{\underline{0}\}\}$.

Lemma 3.1: distance and weight

For any vectors $\underline{v}, \underline{y} \in \mathbb{F}_q^n$, $d(\underline{v}, \underline{y}) = w(\underline{v} - \underline{y})$.

Proof. Indeed, $d(\underline{v}, \underline{y})$ is the number of positions i, $1 \leq i \leq n$, where $v_i \neq y_i$. Obviously, this is the same as the number of positions i where $v_i - y_i \neq 0$. By definition of the weight, this is $w(\underline{v} - \underline{y})$, as claimed. □

Recall that the minimum distance, $d(C)$, of a code C is a very important parameter which tells us how many errors can the code detect and correct in a codeword. The following theorem shows how one can find $d(C)$ if C is linear.

Theorem 3.2: minimum distance equals weight

$d(C) = w(C)$ for a linear code C.

Proof. Take a codevector $\underline{v}$ such that $w(C) = w(\underline{v})$. Observe, $w(\underline{v}) = w(\underline{v} - \underline{0}) = d(\underline{v}, \underline{0})$ but $\underline{v} \neq \underline{0} \in C$ so $w(\underline{v}) \geq d(C)$. We proved that $w(C) \geq d(C)$.

Now take a pair $\underline{y} \neq \underline{z} \in C$ such that $d(\underline{y}, \underline{z}) = d(C)$. Rewrite this as $w(\underline{y} - \underline{z})$. Since C is linear, $\underline{y} - \underline{z} \in C \setminus \{0\}$ so $w(\underline{y} - \underline{z}) \geq w(C)$. We proved that $d(C) \geq w(C)$. □

Remark: in the proof, we twice used that C is linear: first, $\underline{0} \in C$; second, $\underline{y}, \underline{z} \in C$ implies $\underline{y} - \underline{z} \in C$. This condition is essential.

Remark: given a linear code C, one needs to check only $M - 1$ vectors to compute $d(C) = w(C)$. For a non-linear code, one has to check $M(M-1)/2$ pairs of words to compute the minimum distance d.

Here is a non-trivial construction of a linear code.

Example: the zero sum code

For any finite field $\mathbb{F}_q$ and for any $n \geq 1$ we can define the **zero sum code** in $\mathbb{F}_q^n$ as

$$Z = \{(v_1, v_2, \ldots, v_n) \in \mathbb{F}_q^n \mid v_1 + v_2 + \cdots + v_n = 0 \text{ in } \mathbb{F}_q\}.$$

We note that the zero sum code in $\mathbb{F}_q^n$ is a linear code because Z is the set of solutions to the homogeneous linear equation $v_1 + \cdots + v_n = 0$. It is known from linear algebra (and is easy to check directly) that the sum of two vectors satisfying this equation also satisfies this equation, and scaling a vector satisfying this equation again satisfies the equation. In other words, Z is a vector space.

Binary zero sum codes are very common and have a special name.

Example: The binary even weight code E_n

The **binary even weight code of length** n is defined as

$$E_n = \{\underline{v} \in \mathbb{F}_2^n : w(\underline{v}) \text{ is even}\}.$$

Due to the rules of arithmetic in $\mathbb{F}_2$ we have

$$E_n = \{x_1 x_2 \ldots x_n : x_i \in \mathbb{F}_2,\ x_1 + x_2 + \cdots + x_n = 0 \text{ in } \mathbb{F}_2\}$$

which shows that E_n is a particular case of a zero sum code, hence is a linear code.

Note: 0 is an even number! The binary even weight code contains the codeword $00\ldots0$.

Basic properties of the binary even weight code E_n

Minimum distance = weight: a vector with only one 1 has odd weight but a vector $1100\ldots0$ of weight 2 is in E_n. Hence $d(E_n) = w(E_n) = 2$. The code detects up to 1 error and corrects up to 0 errors.

The number of codewords: in a codeword $\underline{v} = (x_1, x_2, \ldots, x_n)$, the first $n-1$ bits can be arbitrary (2^{n-1} combinations), and the last bit is uniquely determined by $x_n = x_1 + \ldots + x_{n-1}$, where $+$ is the addition is in the field $\mathbb{F}_2$. We thus have 2^{n-1} codewords.

Another argument to that effect is as follows. We can take a binary word and flip (change) its first bit. This operation splits the set $\mathbb{F}_2^n$ into pairs of vectors, such that the vectors in a pair only differ in the first bit. Each pair contains one vector of even weight and one vector of odd weight. Therefore, the number of vectors of even weight is equal to the number of vectors of odd weight, and is $\frac{1}{2}\#\mathbb{F}_2^n = 2^{n-1}$.

Conclusion: E_n is an $[n, n-1, 2]_2$-code.

Remark: A widely used code. If an error is *detected*, the recipient will request retransmission of the codeword where the error occurred. Error *correction* is not available.

The code generated by a matrix. A generator matrix of a linear code

We have an unlimited supply of linear codes, due to the following construction.

Definition: the linear code generated by a matrix

Let G be a $k \times n$ matrix with linearly independent rows $\underline{r}_1, \ldots, \underline{r}_k \in \mathbb{F}_q^n$. The code

$$C = \{u_1\underline{r}_1 + \ldots + u_k\underline{r}_k \mid u_1, \ldots, u_k \in \mathbb{F}_q\} \quad \subseteq \quad \mathbb{F}_q^n$$

is said to be **generated by the matrix** G. In this case, the function

$$\texttt{ENCODE}\colon \mathbb{F}_q^k \to C, \qquad \texttt{ENCODE}(\underline{u}) = \underline{u}G \quad \text{for all } \underline{u} \in \mathbb{F}_q^k$$

is the **encoder** for C given by the matrix G.

Proposition 3.3: properties of a code generated by a matrix

In the above definition, C is a linear code. The function ENCODE is a bijective linear map between $\mathbb{F}_q^k$ and C. The **information dimension** of C is k and **is equal to vector space dimension,** $\dim C$.

Proof. The definition says that C is the span of $\underline{r}_1, \ldots, \underline{r}_k$ in the vector space $\mathbb{F}_q^n$. By linear algebra, a span is a subspace of $\mathbb{F}_q^n$ hence a linear code.

Matrix multiplication is linear in each argument so $\texttt{ENCODE}(\underline{u}) = \underline{u}G$ is a linear function of $\underline{u} = (u_1, \ldots, u_k)$. As C consists of vectors of the form $u_1\underline{r}_1 + \cdots + u_k\underline{r}_k = \underline{u}G$, the image of ENCODE is C so ENCODE is surjective. The kernel of ENCODE is made up of all

$(u_1, \ldots, u_k)$ such that $u_1\underline{r}_1 + \ldots + u_k\underline{r}_k = \underline{0}$, but as $\underline{r}_1, \ldots, \underline{r}_k$ are linearly independent, $\ker \texttt{ENCODE} = \{\underline{0}\}$ and so ENCODE is injective, hence bijective.

Hence $M = \#C = \#\mathbb{F}_q^k = q^k$ and so the information dimension of C is $\log_q(M) = k$.

On the other hand, the vector space dimension of C is, by definition, the number of element in a basis of C. Note that the k-element set $\{\underline{r}_1, \ldots, \underline{r}_k\}$ is a basis of C, as this is a linearly independent set which spans C. Hence $\dim C$ is also k. □

In fact, **all** linear codes arise from the above construction. Indeed, we know from linear algebra that every vector space C has a basis. So every linear code is generated by a matrix:

Definition: generator matrix

Let $C \subseteq \mathbb{F}_q^n$ be a linear code. A **generator matrix** of C is a matrix $G = \begin{bmatrix} \underline{r}_1 \\ \underline{r}_2 \\ \vdots \\ \underline{r}_k \end{bmatrix}$, where the row vectors $\underline{r}_1, \ldots, \underline{r}_k$ are a basis of C. (Clearly, C is generated by any of its generator matrices.)

Let us consider some simple matrices and work out the codes they generate.

Example: matrices that can generate a trivial code

The identity matrix I_n is a generator matrix for the trivial code, $\mathbb{F}_q^n$. Any other $n \times n$ matrix with linearly independent rows is also a generator matrix for the trivial code of length n.

Example: matrices that generate repetition codes

The repetition code $Rep(n, \mathbb{F}_q)$ has generator matrix $G = \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}$, of size $1 \times n$. The matrix λG for any $\lambda \in \mathbb{F}_q$, $\lambda \neq 0$ is also a generator matrix for $Rep(n, \mathbb{F}_q)$.

Example: matrices that generate the binary even weight code E_3

$E_3 = \{000, 011, 101, 110\}$ has $4 = 2^2$ codewords, so the dimension of this code is 2. Therefore, a generator matrix has 2 rows and 3 columns.
To write down a generator matrix, we need to take two linearly independent codevectors. We must not use the zero codevector, 000, because a linearly independent

set must not contain the zero vector, but can use any two others. So, each of

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

is a generator matrix for E_3.

Discussion: storing generator matrix instead of the whole code

Thus, to work with a linear code, it is enough to store just its generator matrix instead of storing all codevectors. This approach to linear codes has its practical advantages and disadvantages.

The single **advantage** which outweighs everything else is the amount of storage space required.

> To visualise the difference between storing all the q^k codewords of a linear code and storing only k rows of a generator matrix, consider a binary code of dimension about 1500 used in computer networking for error detection. We can store 1500 rows of a generator matrix, but it is absolutely impossible to store a list of all 2^{1500} codewords. Indeed, the number 10^{100} (the *googol*) is believed to be bigger than the number of electrons in the visible Universe; but googol is less than 2^{340}.

Disadvantages. A generator matrix is in general **not unique**, because a basis of a vector space C can be chosen in more than one way. It may not be obvious if two matrices generate the same code (although it is easy to test by bringing both matrices to reduced row echelon form and comparing the result).

If a linear code C is specified by a generator matrix G, it may be difficult to compute the **weight** $w(C)$ of C. Of course, the weight of C does not exceed, but is in general not equal to, the minimum weight of a row of G. For some linear codes which have been used in practice, the weight is not known!

Generator matrices in standard form

For a linear code C, the encoder, $\texttt{ENCODE}(\underline{u}) = \underline{u}G$, depends on the choice of a generator matrix G. In practice, for many codes there is the best choice:

Definition: matrix in standard form

A matrix G is in *standard form* if its leftmost colums form an identity matrix:

$$G = [I_k \mid A] = \left[\begin{array}{cccc|ccc} 1 & 0 & \dots & 0 & * & \dots & * \\ 0 & 1 & \dots & 0 & * & \dots & * \\ & & \ddots & & & & \\ 0 & 0 & \dots & 1 & * & \dots & * \end{array}\right].$$

Note that entries in the last $n-k$ columns, denoted $*$, are arbitrary elements of $\mathbb{F}_q$.

If G is in standard form, then, after encoding, the first k symbols of the codeword show the original message:

$$\underline{u} \in \mathbb{F}_q^k \quad \mapsto \quad \texttt{ENCODE}(\underline{u}) = \underline{u}G = \underline{u}[I_k \mid A] = [\underline{u} \mid \underline{u}A]$$

(this is an easy example of multiplication of block matrices). This means that it is easy to **unencode** a codevector, simply by taking its first k symbols.

In this situation, the first k symbols of a codeword are called *information symbols*. The last $n-k$ symbols are called *check symbols*; their job is to protect the information from noise by increasing the Hamming distance between codewords.

Theorem 3.4: generator matrix in standard form

If a generator matrix in standard form exists for a linear code C, it is unique, and any generator matrix can be brought to the standard from by the following operations:

(R1) Permutation of rows.

(R2) Multiplication of a row by a non-zero scalar.

(R3) Adding a scalar multiple of one row to another row.

Proof. Not given — a standard fact from linear algebra (uniqueness of reduced row echelon form). We will do examples to show how to find the generator matrix in standard form. □

Remark. If we apply a sequence of the row operations (R1), (R2) and (R3) to a generator matrix of a code C, we again obtain a generator matrix of C. This is implied in the Theorem, and follows from the fact that a basis of a vector space remains a basis under permutations, multiplication of an element of the basis by a scalar, and adding a scalar multiple of an element to another element. This fact is known from linear algebra.

Examples of finding a generator matrix in standard form, and some codes which have no generator matrix in standard form, are on example sheets. We consider one example here:

Example: bringing a generator matrix into standard form

The binary code C is generated by $\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$. Find the generator matrix in standard form for C. Find the parameters of C. Identify the code C by its well-known name.

Solution: apply row operations $\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$ $(r_1 \leftrightarrow r_2)$ $\begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$ $(r_3 \to r_3 + r_1,\, r_4 \to r_4 + r_1)$ $\begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$ $(r_2 \leftrightarrow r_4)$ $\begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}$ $(r_3 \to r_3 + r_2,\, r_4 \to r_4 + r_2)$ $\begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$ $(r_1 \to r_1 + r_4)$ $\begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$ $(r_4 \to r_4 + r_3)$ $\begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$.

The parameters of C are: length 5 (the number of columns of the generator matrix), dimension 4 (the number of rows of the generator matrix). From the generator matrix in standard form (its rows are also codevectors!) we can see that $w(C) \leq 2$. In fact, all the rows of the generator matrix are of even weight; hence they lie in the vector space E_5. Hence all their linear combinations lie in E_5. Since $\dim C = 4 = \dim E_5$, we have $C = E_5$ (*the even weight code of length 5*) and $d(C) = w(C) = 2$.