The Finite Field Z_2

From now on, we look only at binary channels, whose input and output alphabets are both $\{0,1\}$.

We will look at the symbols 0 and 1 as elements of Z_2, the integers considered modulo 2.

Z_2 (also called F_2 or $GF(2)$) is the smallest example of a “field” — a collection of “numbers” that behave like real and complex numbers. Specifically, in a field:

- Addition and multiplication are defined. They are commutative and associative. Multiplication is distributive over addition.
- There are numbers called 0 and 1, such that $z+0=z$ and $z\cdot 1=z$ for all z.
- Subtraction and division (except by 0) can be done, and these operations are the inverses of addition and multiplication.

Arithmetic in Z_2

Addition and multiplication in Z_2 are defined as follows:

$$\begin{array}{ll} 0+0=0 & 0\cdot 0=0 \\ 0+1=1 & 0\cdot 1=0 \\ 1+0=1 & 1\cdot 0=0 \\ 1+1=0 & 1\cdot 1=1 \end{array}$$

This can also be seen as arithmetic modulo 2, in which we always take the remainder of the result after dividing by 2.

Viewed as logical operations, addition is the same as ‘exclusive-or’, and multiplication is the same as ‘and’.

Note: In Z_2, $-a=a$, and hence $a-b=a+b$.

Vector Spaces Over Z_2

Just as we can define vectors over the reals, we can define vectors over any other field, including over Z_2. We get to add such vectors, and multiply them by a scalar from the field.

We can think of these vectors as N-tuples of field elements. For instance, with vectors of length five over Z_2:

$$\begin{aligned} (1,0,0,1,1)+(0,1,0,0,1) &= (1,1,0,1,0) \\ 1\cdot(1,0,0,1,1) &= (1,0,0,1,1) \\ 0\cdot(1,0,0,1,1) &= (0,0,0,0,0) \end{aligned}$$

Most properties of real vector spaces hold for vectors over Z_2 — eg, the existence of basis vectors.

We refer to the vector space of all N-tuples from Z_2 as Z_2^N. We will use boldface letters such as $\mathbf{u}$ and $\mathbf{v}$ to refer to such vectors.

Linear Codes

We can view Z_2^N as the input and output alphabet of the Nth extension of a binary channel.

A code, $\mathcal{C}$, for this extension of the channel is a subset of Z_2^N.

$\mathcal{C}$ is a *linear code* if the following conditions hold:

1) If $\mathbf{u}$ and $\mathbf{v}$ are codewords of $\mathcal{C}$, then $\mathbf{u}+\mathbf{v}$ is also a codeword of $\mathcal{C}$.

2) If $\mathbf{u}$ is a codeword of $\mathcal{C}$ and z is in Z_2, then $z\mathbf{u}$ is also a codeword of $\mathcal{C}$.

In other words, $\mathcal{C}$ must be a subspace of Z_2^N. Note that the all-zero codeword must be in $\mathcal{C}$, since $\mathbf{0}=0\mathbf{u}$ for any $\mathbf{u}$.

Note: For binary codes, condition (2) will always hold if condition (1) does, since $1\mathbf{u}=\mathbf{u}$ and $0\mathbf{u}=\mathbf{0}=\mathbf{u}+\mathbf{u}$.

Linear Codes From Basis Vectors

We can construct a linear code by choosing K linearly-independent *basis vectors* from Z_2^N.

We'll call the basis vectors $\mathbf{u}_1, \ldots, \mathbf{u}_K$. We define the set of codewords to be all those vectors that can be written in the form

$$a_1\mathbf{u}_1 + a_2\mathbf{u}_2 + \cdots + a_K\mathbf{u}_K$$

where $a_1, \ldots, a_K$ are elements of Z_2.

The codewords obtained with different $a_1, \ldots, a_K$ are all different. (Otherwise $\mathbf{u}_1, \ldots, \mathbf{u}_K$ wouldn't be linearly-independent.)

There are therefore 2^K codewords. We can encode a block consisting of K symbols, $a_1, \ldots, a_k$, from Z_2 as a codeword of length N using the formula above.

This is called an $[N, K]$ code. (MacKay's book uses (N, K), but that has another meaning in other books.)

Linear Codes From Linear Equations

Another way to define a linear code for Z_2^N is to provide a set of simultaneous equations that must be satisfied for $\mathbf{v}$ to be a codeword.

These equations have the form $\mathbf{c} \cdot \mathbf{v} = 0$, ie

$$c_1v_1 + c_2v_2 + \cdots + c_Nv_N \;=\; 0$$

The set of solutions is a linear code because

1) $\mathbf{c} \cdot \mathbf{u} = 0$ and $\mathbf{c} \cdot \mathbf{v} = 0$ implies $\mathbf{c} \cdot (\mathbf{u} + \mathbf{v}) = 0$.

2) $\mathbf{c} \cdot \mathbf{v} = 0$ implies $\mathbf{c} \cdot (z\mathbf{v}) = 0$.

If we have $N - K$ such equations, and they are independent, the code will have 2^K codewords.

The Repetition Codes Over Z_2

A repetition code for Z_2^N has only two codewords — one has all 0s, the other all 1s.

This is a linear $[N, 1]$ code, with $(1, \ldots, 1)$ as the basis vector.

The code is also defined by the following $N - 1$ equations satisfied by a codeword $\mathbf{v}$:

$$v_1 + v_2 = 0, \quad v_2 + v_3 = 0, \quad \cdots, \quad v_{N-1} + v_N = 0$$

The Single Parity-Check Codes

An $[N, N-1]$ code over Z_2 can be defined by the following single equation satisfied by a codeword $\mathbf{v}$:

$$v_1 + v_2 + \cdots + v_N \;=\; 0$$

In other words, the *parity* of all the bits in a codeword must be even.

This code can also be defined using $N - 1$ basis vectors. One choice of basis vectors when $N = 5$ is as follows:

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

A $[5, 2]$ *Binary Code*

Recall the following code from lecture 9B:

$$\{\ 00000,\ \ 00111,\ \ 11001,\ \ 11110\ \}$$

Is this a linear code? We need to check that all sums of codewords are also codewords:

$$00111 + 11001 = 11110$$
$$00111 + 11110 = 11001$$
$$11001 + 11110 = 00111$$

We can generate this code using 00111 and 11001 as basis vectors. We then get the four codewords as follows:

$$0 \cdot 00111 + 0 \cdot 11001 = 00000$$
$$0 \cdot 00111 + 1 \cdot 11001 = 11001$$
$$1 \cdot 00111 + 0 \cdot 11001 = 00111$$
$$1 \cdot 00111 + 1 \cdot 11001 = 11110$$

The $[7, 4]$ *Binary Hamming code*

The $[7, 4]$ Hamming code is defined over Z_2 by the following equations that are satisfied by a codeword **u**:

$$\begin{aligned} u_1 + u_2 + u_3 + u_5 &= 0 \\ u_2 + u_3 + u_4 + u_6 &= 0 \\ u_1 + u_3 + u_4 + u_7 &= 0 \end{aligned}$$

Since these equations are independent, there should be 16 codewords.

We can also define the code in terms of the following four basis vectors:

$$1000101,\ \ 0100110,\ \ 0010111,\ \ 0001011$$

There are other sets equations and other sets of basis vectors that define the same code.

We will see later that this code is capable of correcting any single error.