1 Hamming Distance

Throughout this document $\mathbb{F}$ means the binary field $\mathbb{F}_2$.

In $\mathbb{F}_2$ we could define dot product, magnitude and distance in analogy with $\mathbb{R}^n$, but in this case we would get all vectors having length 0 or 1, not very interesting. Instead we use a different definition of magnitude and distance, which is much more useful in this case.

Definition 1 (Hamming distance) *Given two vectors* $\mathbf{u}, \mathbf{v} \in \mathbb{F}^n$ *we define the* hamming distance *between* $\mathbf{u}$ *and* $\mathbf{v}$*,* $d(\mathbf{u}, \mathbf{v})$*, to be the number of places where* $\mathbf{u}$ *and* $\mathbf{v}$ *differ.*

Thus the Hamming distance between two vectors is the number of bits we must change to change one into the other.

Example Find the distance between the vectors 01101010 and 11011011.

$$\begin{array}{c} 01101010 \\ 11011011 \end{array}$$

They differ in four places, so the Hamming distance $d(01101010, 11011011) = 4$.

Definition 2 (Weight) *The* weight *of a vector* $\mathbf{u} \in \mathbb{F}^{\mathbf{n}}$ *is* $w(\mathbf{u}) = d(\mathbf{u}, \mathbf{0})$*, the distance of* $\mathbf{u}$ *to the zero vector.*

The weight of a vector is equal to the number of 1's in it. The weight may be thought of as the magnitude of the vector.

Example Find the weight of 11011011.

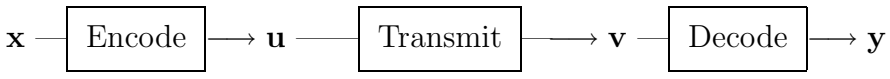
11011011 contains 6 ones, so $w(11011011) = 6$.

2 Error Correcting Codes

Error correcting codes are used in many places, wherever there is the possibility of errors during transmission. Some examples are NASA probes (Galileo), CD players and the Ethernet transmission protocol.

We assume that the original message consists of a series of bits, which can be split into equal size blocks and that each block is of length n, i.e. a member of $\mathbb{F}^n$.

The usual process consists of the original block $\mathbf{x} \in \mathbb{F}^n$ this is then encoded by some *encoding function* to $\mathbf{u} \in \mathbb{F}^{n+k}$ which is then sent across some (noisy) channel. At the other end the received value $\mathbf{v} \in \mathbb{F}^{n+k}$ is decode by means of the corresponding *decoding function* to some $\mathbf{y} \in \mathbb{F}^n$.



If there are no errors in the channel $\mathbf{u} = \mathbf{v}$ and $\mathbf{x} = \mathbf{y}$.

Definition 3 (Code) *A* code *is a set $\mathcal{C} \subset \mathbb{F}^m$, where $m = n + k$, together with a 1-1 encoding transformation $T : \mathbb{F}^n \longrightarrow \mathbb{F}^m$ with $Ran(T) = \mathcal{C}$ and an onto decoding transformation $D : \mathcal{C} \longrightarrow \mathbb{F}^n$.*

In practice the domain of D is often larger than $\mathcal{C}$ to allow for corrections.

Let d be the smallest Hamming distance between two codewords in a code $\mathcal{C}$, $d = \min_{\mathbf{u},\mathbf{v}\in\mathcal{C}}\{d(\mathbf{u}, \mathbf{v})\}$. Thus to change one codeword to another requires at least d bit changes. Then $\mathcal{C}$ can detect up to $d - 1$ errors, since any $d - 1$ transmission errors cannot change one codeword to another.

A code is characterized by the three numbers:
- n - the original message length (bits),
- k - the number of bits added in encoding, and
- d - the minimum distance between codewords.

Suppose that $\mathbf{u}$ was sent and $\mathbf{v}$ was received and $d(\mathbf{u}, \mathbf{v}) \leq (d - 1)/2$, ie. less than $(d - 1)/2$ errors occurred. Then, the distance of $\mathbf{v}$ to any codeword other than $\mathbf{u}$, $\mathbf{w} \in \mathcal{C}$ say, is greater than $(d-1)/2$, since $d(\mathbf{u}, \mathbf{w}) \geq d$ by the definition of d. Thus $\mathbf{u}$ is the nearest codeword to $\mathbf{v}$, the number of bit changes required to get from $\mathbf{u}$ to $\mathbf{v}$ (the number of errors in the channel) is less than the number of errors required to get from any other codeword to $\mathbf{v}$. We correct $\mathbf{v}$ to $\mathbf{u}$, so $\mathcal{C}$ can correct up to $t = (d - 1)/2$ errors.

Example ($3\times$ Repetition Code)
$n = 1$, each bit is a block so a message is either 0 or 1. $k = 2$, so $m = 3$, $\mathcal{C} = \{000, 111\}$.
Encode: (T)
$0 \rightarrow 000$
$1 \rightarrow 111$.
Decode: (D)
$001, 010, 100, 000 \rightarrow 0$
$110, 101, 011, 111 \rightarrow 1$.

$d = 3$, so this code can detect up to 2 errors $(d - 1)$ and correct up to 1 $((d - 1)/2)$.

In general we wish to keep k as low as possible - we want to add as few extra bits as possible, whilst getting d as high as possible, to enable us to detect as many errors as possible. For a given value of d we may measure the efficiency of a code by the information rate $R = n/(n + k)$. The repetition code above has $d = 3$ with information rate $R = 1/3$, the encoded message is three times as long as the original, which is not very good.

Definition 4 (Linear Codes) *A code is called a* linear code *if the transformation T is a matrix transformation. In this case there will be an $(n + k) \times n$ zero-one matrix G such that $T(x) = Gx$, G is called the* generator *of the code.*

Since T_G is $1-1$, G is row equivalent to a matrix with n pivots. In particular it is always possible to reduce the first n rows of G to the $n \times n$ identity matrix, I_n. Thus we will assume that $G = \begin{pmatrix} I_n \\ A \end{pmatrix}$, where A is a $k \times n$ matrix. In this case if the original message is $\mathbf{x}$, then the encoded message $G\mathbf{x} = \begin{pmatrix} \mathbf{x} \\ \mathbf{p} \end{pmatrix}$, where $\mathbf{p} = A\mathbf{x}$. The components of $\mathbf{p}$ are called the *parity bits*, the matrix A is called the *parity matrix* and the equations $A\mathbf{x}$ are called the *parity equations*.

Note that the columns of G are codewords, the first column is the encoding of $10\ldots0$, the second of $010\ldots0$, the third $001\ldots0$, etc. In fact the columns of G form a basis for $\mathcal{C}$.

Example ($(n,k,d)=(4,3,3)$ Hamming code)

This code adds three parity bits to each nibble and corrects up to 1 error. This code has $d=3$ with information rate $R=4/7$, thus the encoded message is $7/4$ times as long as the original, much better than the $3\times$ repetition code above.

$$G=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\\1&1&0&1\\1&0&1&1\\0&1&1&1\end{pmatrix},\ \text{so } A=\begin{pmatrix}1&1&0&1\\1&0&1&1\\0&1&1&1\end{pmatrix}$$

To encode $\mathbf{x}=0110$ we compute $G\mathbf{x}=0110110$, 110 are the parity bits.

3 Parity Check Matrices

We wish to be able to decode messages, check for errors and correct them quickly. One common method is *Syndrome Decoding* which relies on the parity check matrix for a code.

Definition 5 (Parity Check Matrix) *Given a linear code $\mathcal{C}$ with generator G, an $n\times(n+k)$ matrix H is called a* parity check matrix for $\mathcal{C}$ *if $\mathbf{v}\in\mathcal{C}$ if and only if $H\mathbf{v}=\mathbf{0}$.*

Theorem 6 *Given a linear code $\mathcal{C}$ with generator $G=\begin{pmatrix}I_n\\A\end{pmatrix}$, then the corresponding parity check matrix is $H=(A\mid I_k)$. Further $HG=O$, where O is the $n\times n$ zero matrix.*

Proof: Let $\mathcal{C}$, G and H be as given above, we must show that $\mathbf{v}\in\mathcal{C}\Leftrightarrow H\mathbf{v}=\mathbf{0}$.

($\Rightarrow$) Assume $\mathbf{v}\in\mathcal{C}$, since $\mathbf{v}$ is a codeword it must be the encoding of some message $\mathbf{x}\in\mathbb{F}^n$, i.e. $\mathbf{v}=G\mathbf{x}$ for some $\mathbf{x}\in\mathbb{F}^n$. So

$$H\mathbf{v}=HG\mathbf{x}=(A\mid I_k)\begin{pmatrix}I_n\\A\end{pmatrix}\mathbf{x}=(A+A)\mathbf{x}=O\mathbf{x}=\mathbf{0}.$$

(Remember that A is a $k\times n$ zero one matrix and all addition is binary, so $A+A=O$.)
Note that we have also shown that $HG=O$.

($\Leftarrow$) Assume $\mathbf{v}\in\mathbb{F}^{n+k}$ and $H\mathbf{v}=\mathbf{0}$. Write $\mathbf{v}=\begin{pmatrix}\mathbf{v}_1\\\mathbf{v}_2\end{pmatrix}$, where $\mathbf{v}_1\in\mathbb{F}^n$ and $\mathbf{v}_2\in\mathbb{F}^k$. Now

$$\mathbf{0}=H\mathbf{v}=(A\mid I_k)\begin{pmatrix}\mathbf{v}_1\\\mathbf{v}_2\end{pmatrix}=A\mathbf{v}_1+\mathbf{v}_2$$

So $A\mathbf{v}_1 + \mathbf{v}_2 = \mathbf{0}$, i.e. $\mathbf{v}_2 = A\mathbf{v}_1$. Thus

$$\mathbf{v} = \begin{pmatrix} \mathbf{v}_1 \\ A\mathbf{v}_1 \end{pmatrix} = \begin{pmatrix} I_n \\ A \end{pmatrix} \mathbf{v}_1 = G\mathbf{v}_1$$

and so $\mathbf{v} \in \mathcal{C}$.

We can also calculate d from the parity check matrix. If there are d columns of H whose sum is $\mathbf{0}$, but no set of $d-1$ columns of H sum to $\mathbf{0}$ then the code has minimum distance d. i.e. Every set of $d-1$ column vectors from H is linearly independent, but there is some set of d column vectors of H which is linearly dependent.

Multiplication of binary matrices is fast, especially if implemented in hardware. Thus we may quickly detect errors in an incoming transmission $\mathbf{v}$ by computing $H\mathbf{v}$, if it is non-zero an error has occurred. If no error has occurred the original message may be recovered by stripping off the parity bits.

4 Syndrome Decoding

Let $\mathcal{C}$ be a linear code with generator G, parity check matrix H and minimum distance d. Let $t = (d-1)/2$, then we may correct up to t errors. Suppose $\mathbf{u} = G\mathbf{x}$ is sent, and up to t errors occur, then $\mathbf{v} = G\mathbf{x} + \mathbf{e}$ will be received, where $\mathbf{e} \in \mathbb{F}^{n+k}$ has a 1 in each position that was changed in the transmission of $\mathbf{u}$, we are assuming $w(\mathbf{e}) \leq t$. Consider the action of H on the received vector:

$$H\mathbf{v} = H(G\mathbf{x} + \mathbf{e}) = HG\mathbf{x} + H\mathbf{e} = O\mathbf{x} + H\mathbf{e} = H\mathbf{e} = \mathbf{s}.$$

We can calculate $H\mathbf{v}$ on the received vector to get the value $\mathbf{s} = H\mathbf{e}$. We would like to invert the action of H on $\mathbf{s}$ to retrieve $\mathbf{e}$, but H is not invertible (it's not even square). However, we know that $w(\mathbf{e}) \leq t$ and so it is feasible to do the inversion for such vectors by means of a lookup table, called the syndrome table.

A *syndrome* of $\mathbf{e} \in \mathbb{F}^{n+k}$ is $\mathbf{s} = H\mathbf{e} \in \mathbb{F}^k$, if $\mathbf{e}$ is a codeword its syndrome will be $\mathbf{0}$. We keep a table of the syndromes of all $\mathbf{e} \in \mathbb{F}^{n+k}$ with $w(\mathbf{e}) \leq t$, i.e. all $\mathbf{e}$ with less than t ones. If we receive $\mathbf{v}$, we calculate $\mathbf{s} = H\mathbf{v}$, then the corrected codeword will be $\mathbf{v} + \mathbf{e}$, where $\mathbf{e}$ has syndrome $\mathbf{s}$ from the precalculated table.

Examples

1. The $3\times$ repetition code given above is a linear code with $G = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$, so $A = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, and $H = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$, $d = 3$, and $t = 1$. We now calculate the syndrome table, $001, 010, 100$ are the vectors in $\mathbb{F}^3$ with one 1.

$\mathbf{e}$	$\mathbf{s} = H\mathbf{e}$
000	00
001	01
010	10
100	11

Thus if we receive 101, calculate the syndrome $H \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$. 10 corresponds to 010 from the table, so the corrected codeword is $101 + 010 = 111$ and the message was 1.

2. The $(4, 3, 3)$ Hamming code given above has

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}, \text{ so } A = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

so the parity check matrix

$$H = (A \mid I_3) = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

The syndrome table is

$\mathbf{e}$	$\mathbf{s} = H\mathbf{e}$
0000000	000
0000001	001
0000010	010
0000100	100
0001000	111
0010000	011
0100000	101
1000000	110

Thus if we receive 1001100, we calculate the syndrome $H(1001100)^T = (101)^T$, which corresponds to 0100000 from the table, so the corrected codeword is $1001100 + 0100000 = 1101100$, and the message was 1101.

5 Exercises

1. Find the weight of each of the following vectors, find the Hamming distance between the given pairs.

   (a) 0011, 1111 $\in \mathbb{F}^4$

   (b) 0011, 1100 $\in \mathbb{F}^4$

   (c) 11011001, 10011001 $\in \mathbb{F}^8$

   (d) 00111001, 00001001 $\in \mathbb{F}^8$

2. What is the maximum weight of a vector in $\mathbb{F}^n$?
   What is the maximum Hamming distance between two vectors in $\mathbb{F}^n$?

3. A common code is the *Parity Check Code*, in this kind of code one bit is added to each message word, $\mathbf{x} \in \mathbb{F}^n$. This bit is 0 if $w(\mathbf{x})$ is even and 1 if $w(\mathbf{x})$ is odd. The Parity Check Code is a linear code.

   (a) What is d for this code?

   (b) What is the information rate R for this code in terms of n?

   (c) If the original message is 8 bits ($n = 8$) find the generator G for this code. What is the corresponding parity matrix A?

4. The extended Hamming code has generating matrix

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

   (a) Find the encodings of 0011 and 1010.

   (b) i. What is the size of each (unencoded) message (n)?
   ii. How many check bits are added (k)?
   iii. What is the information rate R?
   iv. $d = 3$ for this code, what is the error correction rate t?
   v. Considering your answer to part 4(b)iii, which is the better code, this one or the Hamming code given in the text?

   (c) What is the parity matrix A?

   (d) Find the parity check matrix H.

   (e) Compile the syndrome table for this code.

   (f) In each case below the received vector is given, use H and your table to find what was sent.

   i. 11011100
   ii. 10100111
   iii. 11110100
   iv. 10110101