2 Hamming Distance

Throughout this lecture F is a finite field with q elements.

Definition The *Hamming distance* $d(\underline{x}, \underline{y})$ between two vectors $\underline{x}, \underline{y} \in F^{(n)}$ is the number of coefficients in which they differ, e.g.

$$\begin{array}{ll} \text{in } \mathbb{F}_2^{(5)} & d(00111, 11001) = 4 \\ \text{in } \mathbb{F}_3^{(4)} & d(0122, 1220) = 3. \end{array}$$

Proposition 1 *d satisfies the usual conditions for a metric:*

(a) $d(\underline{x}, \underline{y}) \geq 0$ *and* $d(\underline{x}, \underline{y}) = 0$ *if and only if* $\underline{x} = \underline{y}$

(b) $d(\underline{x}, \underline{y}) = d(\underline{y}, \underline{x})$

(c) $d(\underline{x}, \underline{z}) \leq d(\underline{x}, \underline{y}) + d(\underline{y}, \underline{z})$ *for any* $\underline{x}, \underline{y}, \underline{z} \in F^{(n)}$.

Proof. (a) $d(\underline{x}, \underline{y}) = 0$ if and only if $\underline{x}$, $\underline{y}$ agree in all coordinates and this happens if and only if $\underline{x} = \underline{y}$.

(b) The number of coordinates in which $\underline{x}$ differs from $\underline{y}$ is equal to the number of coordinates in which $\underline{y}$ differs from $\underline{x}$.

(c) $d(\underline{x}, \underline{y})$ is equal to the minimal number of coordinate changes necessary to get from $\underline{x}$ to $\underline{y}$. In its turn, $d(\underline{y}, \underline{z})$ is equal to the minimal number of coordinate changes necessary to get from $\underline{y}$ to $\underline{z}$.

So $d(\underline{x}, \underline{y}) + d(\underline{y}, \underline{z})$ changes will get us from $\underline{x}$ to $\underline{z}$. Hence

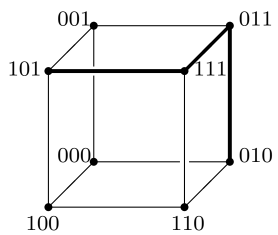
$$d(\underline{x}, \underline{y}) + d(\underline{y}, \underline{z}) \geq d(\underline{x}, \underline{z})$$

which is the minimal number of coordinate changes necessary to get from $\underline{x}$ to $\underline{z}$. $\square$

Example: Hamming distance over the binary alphabet. Words in $\mathbb{F}_2^{(3)}$ can be represented as the vertices

$$000, 001, 010, 011, 100101, 110, 111$$

of a three dimensional cube.



Imagine that the cube is made of wire. Then the Hamming distance between two words is the number of edges in a shortest path connecting the corresponding vertices. For example, $d(101, 010) = 3$. Analogously, the Hamming distance in $\mathbb{F}_2^{(n)}$ can be interpreted as the minimal number of edges in a path connecting two vertices of a n-dimensional cube.

This notion of distance now enables us to make precise the concept of a nearest neighbour.

Nearest neighbour. Given a code $C \subset F^{(n)}$ and a vector $\underline{y} \in F^{(n)}$ then $\underline{x} \in C$ is a *nearest neighbour* to $\underline{y}$ if

$$d(\underline{x}, \underline{y}) = \min\big(d(\underline{z}, \underline{y}) \mid \underline{z} \in C\big)$$

Notice that a vector might have more than one nearest neighbour, so a nearest neighbour is not always unique.

Weight. Let $\underline{v} \in F^n$. Then the *weight* of $\underline{v}$, $w(\underline{v})$, is the number of non-zero co-ordinates in $\underline{v}$.

Lemma *For* $\underline{x}, \underline{y} \in F^n$

$$d(\underline{x}, \underline{y}) = w(\underline{x} - \underline{y}).$$

Proof.

$$\begin{aligned} d(\underline{x}, \underline{y}) &= \text{ number of } \{\, i \mid x_i \neq y_i \,\} \\ &= \text{ number of } \{\, i \mid x_i - y_i \neq 0 \,\} \\ &= w(\underline{x} - \underline{y}). \end{aligned}$$

$\square$

Symmetric channels. Next we consider some of our initial assumptions; these were deliberately omitted in the introduction.

Suppose that a codeword $\underline{x}$ is transmitted and a vector $\underline{y}$ is received. If $\underline{y}$ has a unique nearest neighbour $\underline{x}' \in C$, then it seems "reasonable" to suppose that $\underline{x}'$ was the original message. To justify this we henceforth suppose:

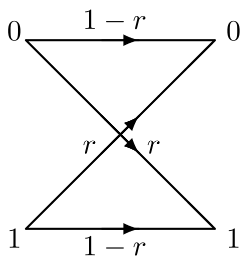
- Errors in different positions in a word are independent; the occurrence of an error in one position in the word does not affect the probability of an error in another position.

- Each symbol $f \in F$ has the same probability r of being erroneously transmitted. We also assume that this probability of error is small, $r \ll 1/2$.

- If $f \in F$ is mistransmitted, then we suppose that all $q - 1$ remaining symbols are equally likely to be received.

We call such channel a *q-ary symmetric channel.*

Example: Binary symmetric channel. Consider the alphabet $F = \mathbf{F}_2$ with the above assumptions and parameters on the channel. Then the above conditions can be summarised schematically by the following diagram:



Suppose now that a q-ary codeword of length n is transmitted. Then

(0) the probability of no errors is $(1-r)^n$,

(1) the probability of 1 error in a specified position is $(1-r)^{n-1}r$,

$\vdots$

(i) the probability of i errors in specified positions is $(1-r)^{n-i}r^i$,

$\vdots$

When exactly i errors occur, the number of possible positions for these errors is $\binom{n}{i}$. Here $\binom{n}{i}$ (read 'n choose i') is a *binomial coefficient*, the number of ways to choose i elements from a set of n elements. Recall that

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}.$$

It is easy to see that

$$\binom{n}{1} = n, \binom{n}{2} = \frac{n(n-1)}{2}, \ldots$$

Hence

(0) the probability of no errors is $(1-r)^n$,

(1) the probability of exactly 1 error (in any position) is $n \cdot (1-r)^{n-1}r$;

(2) probability of exactly 2 errors (in arbitrary positions) is

$$\frac{n(n-1)}{2} \cdot (1-r)^{n-2}r^2,$$

$\vdots$

(i) the probability of exactly i errors (in any positions) is

$$\binom{n}{i}(1-r)^{n-i}r^i,$$

$\vdots$

Comparing these probabilities, we see that if r is sufficiently small ($r < 1/(n+1)$ works), then the vector with *no* error is the most likely of these classes to be received. A vector with exactly 1 error is next most likely, etc. We skip the technical details and restrict ourselves to comparing the probabilities of 0 and 1 errors:

$$(1-r)^n > n \cdot (1-r)^{n-1}r$$

is, after cancelation, equivalent to

$$1 - r > n \cdot r,$$

which is equivalent to $r < 1/(n+1)$.

Thus this argument justifies our initial treatment—at least for symmetric channels.

Notice also that even with higher possibility of error, provided that $r < (q-1)/q$, a codeword closest in terms of Hamming distance to the received word has the greatest probability of being the sent codeword. After all, the probability that the received vector is a *specific* word at a distance m from the sent word is $(1-r)^{n-m}(r/(q-1))^m$, which is strictly decreasing with m whenever $r < (q-1)/q$.

The minimum distance. We return now to the Hamming distance function. In order to avoid trivialities, in the sequel we *always* suppose that $|C|$ (the number of codewords) is greater than 1.

Definition. The *minimum distance* of a code C, denoted $d(C)$, is

$$d(C) = \min\big(d(\underline{x}, \underline{y}) \mid \underline{x}, \underline{y} \in C, \underline{x} \neq \underline{y}\big)$$

Note that this definition makes sense, since $|C| > 1$. Moreover, it is clear that $d(C) \geq 1$.

Example. Consider $C_1 = \{00, 01, 10, 11\}$. Then $d(C_1) = 1$.

$d(C)$ is a crucial invariant of a code, as is shown by the following simple but very important result.

Theorem 2 ***(a)*** *If, for a code C,*

$$d(C) \geq s + 1$$

then C can detect up to s errors.

(b) *If*

$$d(C) \geq 2t + 1$$

then the code C can correct up to t errors.

Proof.

(a) Suppose that $d(C) \geq s+1$. Let $\underline{x}$ be the codeword sent and $\underline{y}$ the vector received. Suppose that $\underline{y}$ is subject to at most s errors i.e. $d(\underline{x}, \underline{y}) \leq s$. Then $\underline{y}$ cannot be a codeword (since $d(\underline{x}, \underline{y}) < d(C)$); thus the error is detected.

(b) Suppose that $d(C) \geq 2t+1$. Let $\underline{x}$ be the codeword sent and $\underline{y}$ be the vector received. Suppose that $\underline{y}$ has at most t errors, i.e. $d(\underline{x}, \underline{y}) \leq t$.

If $\underline{x}'$ is any codeword different from $\underline{x}$, then we claim that $d(\underline{x}', \underline{y}) \geq t+1$, because

$$\begin{aligned} d(\underline{x}', \underline{y}) + t \geq d(\underline{x}', \underline{y}) + d(\underline{y}, \underline{x}) &\geq d(\underline{x}', \underline{x}) \\ &\geq d(C) \\ &\geq 2t+1. \end{aligned}$$

This means that $\underline{x}$ is the unique nearest neighbour to $\underline{y}$, so $\underline{y}$ may be corrected to $\underline{x}$ by finding the nearest neighbour to y and choosing that as the decoded word.

This completes the proof. □

This method of decoding is called *nearest neighbour decoding*.