More on Hamming Distance

Recall that the Hamming distance, $d(\mathbf{u}, \mathbf{v})$, of two codewords **u** and **v** is the number of positions where **u** and **v** have different symbols.

This is a proper distance, which satisfies the *triangle inequality*:

$$d(\mathbf{u}, \mathbf{w}) \;\leq\; d(\mathbf{u}, \mathbf{v}) + d(\mathbf{v}, \mathbf{w})$$

Here's a picture showing why:

```
u : X X X A A A Q Q Q Q I I
                - - - - - -
v : X X X A A A P P P P J J
          - - -         - -
w : X X X B B B P P P P I I
```

Here, $d(\mathbf{u}, \mathbf{v}) = 6$, $d(\mathbf{u}, \mathbf{v} = 5)$, and $d(\mathbf{u}, \mathbf{w}) = 7$.

Minimum Distance and Decoding

A code's *minimum distance* is the minimum of $d(\mathbf{u}, \mathbf{v})$ over all distinct codewords **u** and **v**.

If the minimum distance is at least $2t + 1$, a nearest neighbor decoder will always decode correctly when there are t or fewer errors.

Here's why: Suppose the code has distance $d \geq 2t + 1$. If **u** is sent and **v** is received, having no more than t errors, then

- $d(\mathbf{u}, \mathbf{v}) \leq t$.
- $d(\mathbf{u}, \mathbf{u}') \geq d$ for any codeword $\mathbf{u}' \neq \mathbf{u}$.

From the triangle inequality:

$$d(\mathbf{u}, \mathbf{u}') \leq d(\mathbf{u}, \mathbf{v}) + d(\mathbf{v}, \mathbf{u}')$$

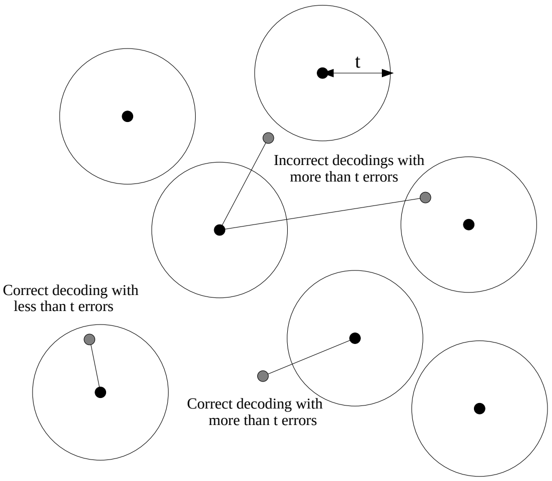
It follows that

$$d(\mathbf{v}, \mathbf{u}') \geq d(\mathbf{u}, \mathbf{u}') - d(\mathbf{u}, \mathbf{v}) \geq d - t \geq (2t+1) - t \geq t+1$$

The decoder will therefore decode correctly to **u**, at distance t, rather than to some other $\mathbf{u}'$.

A Picture of Distance and Decoding

Here's a picture of codewords (black dots) for a code with minimum distance $2t + 1$, showing how some transmissions are decoded:


t
Incorrect decodings with more than t errors
Correct decoding with less than t errors
Correct decoding with more than t errors

Hamming's Sphere-Packing Bound

We'd like to make the minimum distance as large as possible, or alternatively, have as many codewords as possible for a given distance. There's a limit, however.

Consider a binary code with $d = 3$, which can correct any single error. The "spheres" of radius one around each codeword must be disjoint — so that any single error leaves us closest to the correct decoding.

For codewords of length n, each such sphere contains $1+n$ points. If we have M codewords, the total number of points in all spheres will be $M\,(1+n)$, which can't be greater than the total number of points, 2^n.

So for binary codes that can correct any single error, the number of codewords is limited by

$$M \;\leq\; 2^n/(1+n)$$

A More General Version of the Bound

A binary code of length n that is guaranteed to correct any pattern of up to t errors can't have any more than the following number of codewords:

$$2^n \left(1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{t}\right)^{-1}$$

The kth term in the brackets is the number of possible patterns of k errors in n bits:

$$\binom{n}{k} \;=\; \frac{n!}{k!\,(n-k)!}$$

If the above bound is actually reached, the code is said to be *perfect*. For a perfect code, the disjoint spheres of radius t around codewords cover all points.

Very few perfect codes are known. Usually, we can't find a code with as many codewords as would be allowed by this bound.

The Gilbert-Varshamov Bound

The sphere-packing bound is an *upper* limit on how many codewords we can have. There's also a *lower* limit, showing there **is** a code with at least a certain number of codewords.

There is a binary code of length n with minimum distance d that has at least the following number of codewords:

$$2^n \left(1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{d-1}\right)^{-1}$$

Why? Imagine spheres of radius $d-1$ around codewords in a code with fewer codewords than this. The number of points in each sphere is the sum above in brackets, so the total number of points in these spheres is less than 2^n. So there's a point outside these spheres where we could add a codeword that is at least d away from any other codeword.

Minimum Distance for Linear Codes

To find the minimum distance for a code with M codewords, we will in general have to look at all $M(M-1)/2$ pairs of codewords.

But there's a short-cut for linear codes...

Suppose two distinct codewords $\mathbf{u}$ and $\mathbf{v}$ are a distance d apart. Then the codeword $\mathbf{u}-\mathbf{v}$ will have d non-zero elements. The number of non-zero elements in a codeword is called its *weight*.

Conversely, if a non-zero codeword $\mathbf{u}$ has weight d, then the minimum distance for the code is at least d, since $\mathbf{0}$ is a codeword, and $d(\mathbf{u},\mathbf{0})$ is equal to the weight of $\mathbf{u}$.

So the minimum distance of a linear code is equal to the minimum weight of the $M-1$ non-zero codewords.

Examples of Minimum Distance and Error Correction for Linear Codes

Recall the $[5,2]$ code with the following codewords:

00000 00111 11001 11110

The three non-zero codewords have weights of 3, 3, and 4. This code therefore has minimum distance 3, and can correct any single error.
Is this a "perfect" code?

The single-parity check code with $n=4$ has the following codewords:

0000 0011 0101 0110
1001 1010 1100 1111

The smallest weight of a non-zero codeword above is 2, so this is the minimum distance of this code. This is too small to guarantee correction of even one error. (Though the presence of a single error can be detected.)

Finding Minimum Distance From a Parity-Check Matrix

We can find the minimum distance of a linear code from a parity-check matrix for it, H.

The minimum distance is equal to the smallest number of linearly-dependent columns of H.

Why? A vector $\mathbf{u}$ is a codeword iff $\mathbf{u}H^T = \mathbf{0}$. If d columns of H are linearly dependent, let $\mathbf{u}$ have 1s in those positions, and 0s elsewhere. This $\mathbf{u}$ is a codeword of weight d. And if there were any codeword of weight less than d, the 1s in that codeword would identify a set of less than d linearly-dependent columns of H.

Special cases:

- If H has a column of all zeros, then $d = 1$.
- If H has two identical columns, then $d \leq 2$.
- For binary codes, if all columns are distinct and non-zero, then $d \geq 3$.

Example: The [7, 4] *Hamming Code*

We can define the [7, 4] Hamming code by the following parity-check matrix:

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

Clearly, all the columns of H are non-zero, and they are all distinct. So $d \geq 3$. We can see that $d = 3$ by noting that the first three columns are linearly dependent, since

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

This produces 1110000 as an example of a codeword of weight three.

Since it has minimum distance 3, this code can correct any single error. Is it a perfect code?