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:

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

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

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

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.

Hamming Codes

We have seen that a binary $[N, K]$ code will correct any single error if all the columns in its parity-check matrix are non-zero and distinct.

One way to achieve this: Make the $N - K$ bits in successive columns be the binary representations of the integers 1, 2, 3, etc.

This is one way to get a parity-check matrix for a [7, 4] Hamming code:

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

When N is a power of two minus one, the columns of H contain binary representations of all non-zero integers up to $2^{N-K} - 1$.

These are the called the Hamming codes.

Encoding Hamming Codes

By rearranging columns, we can put the parity-check matrix for a Hamming code in systematic form. For the [7, 4] code, we get

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

Recall that a systematic parity check matrix of the form $[P^T \,|\, I_{N-K}]$ goes with a systematic generator matrix of the form $[I_K \,|\, P]$. We get

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

We encode a message block, $\mathbf{s}$, of four bits, by computing $\mathbf{t} = \mathbf{s}G$. The first four bits of $\mathbf{t}$ are the same as $\mathbf{s}$; the remaining three bits are "check bits". Note: The order of bits may vary depending on how the code is constructed.

Decoding Hamming Codes

Consider the non-systematic parity-check matrix:

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

Suppose $\mathbf{t}$ is sent, but $\mathbf{r} = \mathbf{t} + \mathbf{n}$ is received. The receiver can compute the *syndrome* for $\mathbf{r}$:

$$\mathbf{z} = \mathbf{r}H^T = (\mathbf{t}+\mathbf{n})H^T = \mathbf{t}H^T + \mathbf{n}H^T = \mathbf{n}H^T$$

Note that $\mathbf{t}H^T = \mathbf{0}$ since $\mathbf{t}$ is a codeword.

If there were no errors, $\mathbf{n} = \mathbf{0}$, so $\mathbf{z} = \mathbf{0}$.

If there is one error, in position i, then $\mathbf{n}H^T$ will be the ith column of H — which contains the binary representation of the number i!

So to decode, we compute the syndrome, and if it is non-zero, we flip the bit it identifies.

If we rearranged H to systematic form, we modify this procedure in corresponding fashion.

Syndrome Decoding in General

For any linear code with parity-check matrix H, we can find the nearest-neighbor decoding of a received block, $\mathbf{r}$, using the syndrome, $\mathbf{z} = \mathbf{r}H^T$.

We write the received data as $\mathbf{r} = \mathbf{t} + \mathbf{n}$, where $\mathbf{t}$ is the transmitted codeword, and $\mathbf{n}$ is the *error pattern*, so that $\mathbf{z} = \mathbf{n}H^T$.

A nearest-neighbor decoding can be found by finding an error pattern, $\mathbf{n}$, that produces the observed syndrome, and which has the smallest possible weight. Then we decode $\mathbf{r}$ as $\mathbf{r} - \mathbf{n}$.

Building a Syndrome Decoding Table

We can build a table indexed by the syndrome that gives the error pattern of minimum weight for each syndrome.

We initialize all entries in the table to be empty.

We then consider the non-zero error patterns, $\mathbf{n}$, in some order of non-decreasing weight. For each $\mathbf{n}$, we compute the syndrome, $\mathbf{z} = \mathbf{n}H^T$, and store $\mathbf{n}$ in the entry indexed by $\mathbf{z}$, *provided* this entry is currently empty. We stop when the table has no empty entries.

Problem: The size of the table is exponential in the number of check bits — it has $2^{N-K} - 1$ entries for an $[N, K]$ code.

Example: The $[5, 2]$ *Code*

Recall the $[5, 2]$ code with this parity-check matrix:

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

Here is a syndrome decoding table for this code:

z	**n**
001	00001
010	00010
011	00100
100	01000
101	10000
110	10100
111	01100

The last two entries are not unique.

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 more than this 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.

Hamming Codes are Perfect

For each positive integer c, there is a binary Hamming code of length $N = 2^c - 1$ and dimension $K = N - c$. These codes all have minimum distance 3, and hence can correct any single error.

They are also perfect, since

$$2^N/(1+N) \;=\; 2^{2^c-1}/(1+2^c-1) \;=\; 2^{2^c-1-c} \;=\; 2^K$$

which is the number of codewords.

One consequence: A Hamming code can correct any single error, but if there is more than one error, it will not be able to give any indication of a problem — instead, it will "correct" the wrong bit, making things worse.

The *extended Hamming codes* add one more check bit (ie, they have one more row of all 1s to the parity-check matrix). This allows them to detect when two errors have occurred.

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.