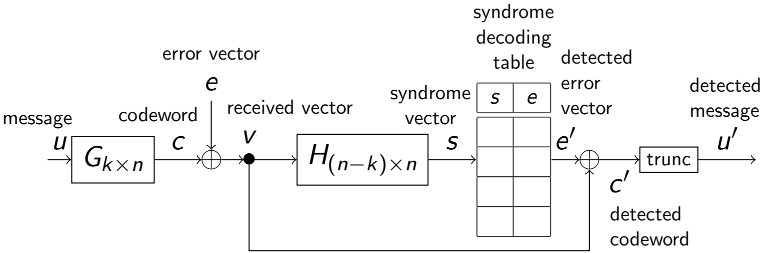
Hamming Codes, Hadamard Codes, Basic code operations

Coding Technology

Illés Horváth

2025/09/24

The binary linear coding scheme



Recall: syndrome decoding, error groups, group leader.

Example: 3× repeater code

For the 3× repeater code, code parameters are $n = 3, k = 1$.

Codewords: $c^{(1)} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \quad c^{(2)} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}.$

Generator matrix G and parity check matrix H:

$$G = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \qquad H = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Syndrome vectors with corresponding error groups and group leaders:

$$\begin{array}{lcl} (00) & \rightarrow & \{(000), (111)\} \\ (01) & \rightarrow & \{(001), (110)\} \\ (10) & \rightarrow & \{(010), (101)\} \\ (11) & \rightarrow & \{(100), (011)\} \end{array}$$

Hamming codes

We aim to construct binary linear codes which are perfect and can correct $t = 1$ error. Such codes are called Hamming codes.

First, let's compute the possible values of n and k. Since the code must be perfect, there is equality in the Hamming bound with $t = 1$:

$$\sum_{i=0}^{t} \binom{n}{i} = 2^{n-k} \qquad\qquad 1 + n = 2^{n-k}.$$

$n + 1$ is a power of 2, so the possible values for n are $3, 7, 15, 31, 63, 127, 255, \ldots$, and the possible (n, k) pairs are:

$(3, 1)$, $(7, 4)$, $(15, 11)$, $(31, 26)$, $(63, 57)$, $(127, 120)$, $(255, 247), \ldots$

There are no Hamming codes for other parameters.

The columns of H and error correction

But first, an important result.

Theorem

Let H denote the parity check matrix of any binary linear code. Then the code can correct ≥ 1 error $\iff$ the column vectors of H are all different and nonzero.

Proof.

The error vector $\begin{bmatrix}0 & \dots & 0\end{bmatrix} \in 2^n$ corresponds to the syndrome $\begin{bmatrix}0 & \dots & 0\end{bmatrix} \in 2^{n-k}$.

The error vector $e^{(1)} = \begin{bmatrix}1 & 0 & \dots & 0\end{bmatrix}$ corresponds to the syndrome which is the first column of H (transposed) due to $s = eH^T$. $e^{(2)} = \begin{bmatrix}0 & 1 & 0 & \dots & 0\end{bmatrix}$ corresponds to the syndrome which is the second column of H, and so on.

The code can correct ≥ 1 error $\iff$ all error vectors of weight 0 and 1 are decoded correctly $\iff$ all error vectors of weight 0 and 1 are in different error groups $\iff$ all error vectors of weight 0 and 1 give a different syndrome vector $\iff$ the columns of H are all different from each other and also the 0 vector.

Hamming codes – construction

For an (n, k) pair from the above list, the $C(n, k)$ Hamming code is constructed the following way.

List all nonzero vectors from $\{0, 1\}^{n-k}$. (How many vectors?) The columns of the parity check matrix H are these vectors, ordered so that the rightmost $(n-k) \times (n-k)$ block of H is the identity matrix. (Why?) The rest of the columns can be in any order.

Example. The parity check matrix of the $C(3, 1)$ Hamming code is

$$H = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}.$$

What is the corresponding generator matrix?

$$G = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}.$$

(What is this code?)

Hamming codes

Example. The parity check matrix of the $C(7,4)$ Hamming code is

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

The corresponding generator matrix can be obtained as

$$H = \begin{array}{c} \left[\begin{array}{|cccc|ccc|} \hline 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ \hline \end{array}\right] \\ \begin{array}{cc} \qquad B \qquad\qquad & \qquad I \qquad \end{array} \end{array} \rightarrow G = \begin{array}{c} \left[\begin{array}{|cccc|ccc|} \hline 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline \end{array}\right] \\ \begin{array}{cc} \qquad I \qquad\qquad & \qquad B^T \qquad \end{array} \end{array}$$

(We have seen this code during the last problem solving session.)

Hamming codes

Due to the earlier theorem, this code can correct ≥ 1 error.

Theorem

For the code constructed above, $d_{\min} = 3$.

Proof. The code can correct ≥ 1 error, so $d_{\min} \geq 3$.

To see that $d_{\min} \leq 3$, consider that

$$H = \left[\begin{array}{c|c} B & I \end{array}\right] \qquad \rightarrow \qquad G = \left[\begin{array}{c|c} I & B^T \end{array}\right],$$

where the columns of B contain all vectors from $\{0,1\}^{n-k}$ with weight 2 or more, so the rows of B^T also contain all vectors from $\{0,1\}^{n-k}$ with weight 2 or more.

Pick any row from B^T with weight 2; that row of G has weight 3 (as all rows of I contain a single 1), which shows

$$d_{\min} = \min_{c \neq (00...0)} w(c) = 3,$$

and the code can correct 1 error and is perfect.

Application to QoS

Hamming codes are useful for good quality channels where the bit error probability is low enough that we expect the typical error situation to be a single error.

In such cases, using the proper Hamming code can decrease the error probability even further, while providing a good code rate.

Example. Assume a channel has bit error probability $p_b = 0.001$. We want to transmit a message of 26 bits. Without error correction,

$$P(\text{decoding error}) = 1 - (1 - 0.001)^{26} = 0.0257.$$

Using a $C(31, 26)$ Hamming code,

$$P(\text{decoding error}) = 1 - (1 - 0.001)^{31} - 31 \cdot 0.001(1 - 0.001)^{30} = 0.000456.$$

Summary for Hamming codes

- Possible parameters: $C(n, k)$, where $n + 1 = 2^{n-k}$, so $(3, 1)$, $(7, 4)$, $(15, 11)$, $(31, 26)$, $(63, 57)$, $(127, 120)$, $(255, 247)$, $\dots$
- Construction: the columns of the parity check matrix H are all different nonzero vectors of length $n - k$.
- $d_{\text{min}} = 3$
- Can detect 2 errors.
- Can correct 1 error.
- Relatively high code rate.
- Useful for good quality channels to reduce the probability of decoding error even further.

Error detection and/or error correction?

We know that any code with $d_{\min} = 3$ can...

- detect 2 errors, and
- correct 1 error.

Detecting 2 errors means that assuming at most 2 errors occurred, the receiver can tell whether the number of errors was 0 or not. However, they cannot distinguish between 1 and 2 errors.

Meanwhile, correcting 1 error means that assuming at most 1 error occurred, the receiver can decode correctly. But if 2 errors occurred, decoding will give a wrong result (since the code is perfect).

We look to improve on this next.

Extended Hamming codes

The main idea is that we want to extend Hamming codes by 1 bit to increase d_{min} to 4; then, assuming at most 2 errors occurred, the receiver can distinguish between the following cases:

- 0 errors occurred, and also decode correctly;
- 1 error occurred, and also decode correctly;
- 2 errors occurred, but cannot tell which two, and cannot decode.

The extra bit is referred to as an additional parity bit.

The extended Hamming codes are no longer perfect.

To distinguish between the two types of Hamming codes, they are referred to as perfect Hamming code and extended Hamming code respectively. (Hamming code usually refers to a perfect Hamming code.)

Extended Hamming codes – construction

Let G denote the generator matrix of a $C(n,k)$ perfect Hamming code. The generator matrix G' of the corresponding extended Hamming code is

$$G' = \left[\ G \mid g\ \right], \quad \text{where} \quad g = G \cdot \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}.$$

The extended Hamming code is $C(n+1,k)$.

Another way to distinguish between perfect and extended Hamming codes is the $C(n,k)$ designations:

- $C(3,1), C(7,4), C(15,11)\ldots$ are perfect Hamming codes;
- $C(4,1), C(8,4), C(16,11)\ldots$ are extended Hamming codes.

Extended Hamming codes

Example. The generator matrix G' of the $C(8,4)$ extended Hamming code is

$$G' = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

and the parity check matrix is

$$H' = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Extended Hamming codes

Theorem

For the extended Hamming code, $d_{\min} = 4$.

Proof. $d_{\min}$ can either be 3 or 4 since adding one extra bit to the codewords changes the distance between codewords by 0 or $+1$.

The weight of each row of G' is even, so any linear combination of rows of G' has even weight too, so the weight of any codeword is also even, so $\min_{c \neq (00\ldots0)} w(c) = d_{\min}$ must also be even, so $d_{\min} = 4$.

(This also implies that all nonzero codewords have weight at least 4.)

Extended Hamming codes

For the extended Hamming code, there are twice as many error groups (2^{n+1-k}). Error vectors with weight 0 or 1 still go into different error groups, but they cover only $2^{n-k} + 1$ groups.

These error groups correspond to syndrome vectors which are equal to one of the columns of H'.

For each of the remaining error groups, the minimal weight is 2, and each of these groups contain a tie for the group leader.

Extended Hamming codes

Example. For the $C(3,1)$ Hamming code,

$$G = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}.$$

For the corresponding $C(4,1)$ extended Hamming code,

$$G' = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}, \qquad H' = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

The codewords are $\begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$ and $\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$.

The error groups are

$$\begin{array}{ccc}
(000) & \rightarrow & \{(0000),(1111)\} \\
(001) & \rightarrow & \{(0001),(1110)\} \\
(010) & \rightarrow & \{(0010),(1101)\} \\
(100) & \rightarrow & \{(0100),(1011)\} \\
(101) & \rightarrow & \{(0101),(1010)\} \\
(110) & \rightarrow & \{(1001),(0110)\} \\
(011) & \rightarrow & \{(0011),(1100)\} \\
(111) & \rightarrow & \{(1000),(0111)\}
\end{array}$$

Extended Hamming codes

Based on this, we can distinguish between 0, 1 and 2 errors the following way:

- If the syndrome vector is the all 0 vector, then 0 errors occurred.
- If the syndrome vector is equal to a column of H' (transposed), then 1 error occurred, and the position of that error is the same as the position of the matching column in H'.
- For any other syndrome vector, 2 errors occurred, but we cannot tell which two.

In the first two cases, we obtain e' and we can proceed with the decoding. For the last case, we do not decode.

This property is called SECDED (Single Error Correction, Double Error Detection).

Extended Hamming codes are preferred to perfect Hamming codes when the channel is noisier (so even double errors may occur), but there is an option to retransmit the codeword.

Summary for extended Hamming codes

- Possible parameters: $C(n+1, k)$, where $n+1 = 2^{n-k}$, so $(4,1)$, $(8,4)$, $(16,11)$, $(32,26)$, $(64,57)$, $(128,120)$, $(256,247)$, $\ldots$
- Construction: generator matrix G' is obtained from the generator G of the (perfect) Hamming code by adding a parity bit.
- $d_{\text{min}} = 4$
- Can correct 1 error and distinguish between 1 and 2 errors (SECDED).
- Slightly worse code rate than perfect Hamming codes.
- Useful for channels which are a little noisier but there is an option to retransmit.

Adding a parity bit

We are going to list a few basic operations that can be used to modify codes.

We have already seen the addition of a parity bit (used to modify Hamming codes into extended Hamming codes).

It works the same in general: if G is the generator matrix of a $C(n, k)$ code, then the generator matrix

$$G' = \left[\; G \; \middle| \; g \; \right], \quad \text{where} \quad g = G \cdot \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}.$$

corresponds to a code with an additional parity bit.

Example:

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix} \quad \rightarrow \quad G' = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

Parity check bit

Another example is the single parity check code (which we have already seen in Lecture 2): we add a single parity bit directly to the message. This results in a $C(k+1, k)$ code with generator

$$G = \begin{bmatrix} 1 & 0 & \dots & 0 & 1 \\ 0 & 1 & \dots & 0 & 1 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & 1 & 1 \end{bmatrix}.$$

$d_{\text{min}} = 2$ for this code, so the code can detect 1 error but cannot correct it.

Adding a parity bit

In general, adding a parity bit ensures that for the code with generator matrix G', every codeword has even weight.

This can be useful if, for the original code, $d_{\text{min}} = \min_{c \neq 0} w(c)$ is odd, because then adding a parity bit increases the minimal code distance by 1.

Essentially, this is what allowed the SECDED property for extended Hamming codes, or the single error detection property of the single parity check code.

What if we add another parity bit to G'?

$$G' = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \quad \rightarrow \quad G'' = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

It does not do anything useful; basically, we added a 0 bit to each codeword that carries no extra information. We might as well remove it. We look at this next.

Punctured code

Puncturing a code means removing one bit from the codewords.

For linear codes, this corresponds to removing a column from the generator matrix G.

Puncturing decreases n by 1, leaves k unchanged, and may change $\min_{c \neq 0} w(c)$ by either 0 or -1:

- If there is a codeword with minimal weight with a 1 bit at the punctured position, then $\min_{c \neq 0} w(c)$ changes by -1;
- otherwise, $\min_{c \neq 0} w(c)$ remains unchanged.

Code puncturing is often used to shorten codewords to a specific length (depending on the application).

Punctured code

If we first add a parity bit to a code, then puncture the last bit, we get back the original code.

What if we first puncture the last bit of a code, then add a parity bit?

In this case, the code can be different from the original:

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

Equivalent codes

A $C(n,k)$ binary block code is equivalent to a $C(n',k')$ code if $n=n', k=k'$ and the bits of the $C(n,k)$ codewords can be rearranged so that the two sets of codewords are the same.

Error correction and error detection capabilities of equivalent codes are the same.

Example. Hamming codes of the same (n,k) parameters are equivalent.

Theorem

Binary linear codes generated by G and G' respectively are equivalent if G and G' have the same size, and G' can be obtained from G by a finite sequence of the following operations:

- *permutation of the rows;*
- *permutation of the columns;*
- *adding one row to another row.*

Dual codes

The dual code of a $C(n, k)$ binary linear code with generator matrix G and parity check matrix H is a $C(n, n-k)$ binary linear code with generator matrix H and parity check matrix G.

The code rate of the dual code is $1 - \frac{k}{n}$, so in general the dual of a code with high code rate has low code rate and vice versa. However, there is no general results connecting d_{min} for the original and dual code.

Example. What is the dual of the $n\times$ repeater code? It is equivalent to the single parity check code.

Codes that are equivalent to their dual are called self-dual.

Example: the $C(8, 4)$ extended Hamming-code is self-dual.

Hadamard codes

Example. What is the dual of the $C(7,4)$ Hamming code? For the dual code,

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Is this a meaningful code?

Hadamard codes are $C(2^k, k)$ codes with $d_{\min} = 2^{k-1}$, so possible parameters are

$$(4,2), (8,3), (16,4), (32,5), (64,6), (128,7), (256,8), \ldots$$

Hamming codes have good code rate and limited error correction capabilities. Repeater codes are at the other extreme: the $n\times$ repeater code has code rate $1/n$ and can correct $\lfloor (n-1)/2 \rfloor$ errors.

Hadamard codes are similar to repeater codes (low code rate, high error correction capabilities), but are better structured.

Hadamard codes – construction

Hadamard codes are constructed the following way. The columns of the generator matrix G are all different binary vectors of length k, in lexicographic order.

Example. For $k = 3$,

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

This is the usual ordering, even though the code is non-systematic.

Hadamard codes are equivalent to the dual code of Hamming codes, with an all-0 column added.

Hadamard codes

Theorem

For the code constructed above, $d(c, c') = 2^{k-1}$ for any $c \neq c'$ codewords.

Proof. Let $u \neq u' \in \{0, 1\}^k$ be two messages, and $c = uG, c' = u'G$ the corresponding codewords. Then

$$d(c, c') = \sum_{g \in \{0,1\}^k} (gu - gu' \mod 2) = \sum_{g \in \{0,1\}^k} g \cdot (u - u')^T.$$

$g \cdot (u - u')^T$ can be either 0 or 1, depending on whether g is orthogonal to $u - u'$ or not. Since $u \neq u'$, u and u' differ in at least 1 position i.

We arrange the total 2^k choices for g into 2^{k-1} pairs which differ only in position i; from each pair $(g^{(1)}, g^{(2)})$, one will give $g^{(1)} \cdot (u - u')^T = 1$ and the other $g^{(2)} \cdot (u - u')^T = 0$. Overall, out of the 2^{k-1} pairs, one from each pair will contribute 1 to $d(c, c')$, so

$$d(c, c') = 2^{k-1}.$$

Hadamard codes

It follows directly that $d_{\text{min}} = 2^{k-1}$, so Hadamard codes can correct

$$\lfloor (d_{\text{min}} - 1)/2 \rfloor = 2^{k-2} - 1$$

errors.

Let's look at the generator for $k = 3$ again.

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

G is obviously not optimal, e.g. the all-0 column could be punctured. But we can do even better than that.

The vector $(11\ldots1) \in \{0,1\}^n$ has Hamming-distance 2^{k-1} from all of the codewords, so it could be added to the list of codewords with d_{min} unchanged. And it is not the only one, e.g. (11110000) or (11001100) could also be added.

Augmented Hadamard codes

Essentially, the current set of codewords uses only half of the codeword space!

In the Hadamard code, $w(c) = 2^{k-1}$ for every nonzero codeword c, so the vector $(11\ldots1) - c$ also has weight 2^{k-1}.

Since this is a linear code, we can add all of these vectors in one step by adding the vector $(11\ldots1)$ as an extra row to G:

$$G' = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

This is called the $C(8, 4)$ augmented Hadamard code.

Augmented Hadamard codes

In general, the augmented Hadamard code $C(2^k, k+1)$ is constructed from the $C(2^k, k)$ Hadamard code by adding an all-1 row to the generator matrix.

Another, equivalent way to obtain G' is to start from the generator matrix of the $C(2^{k+1}, k)$ Hadamard code, and restrict the matrix to columns whose first coordinate is 1.

Augmented Hadamard codes are $C(2^k, k+1)$ codes with $d_{\min} = 2^{k-1}$, so augmented Hadamard codes can correct

$$\lfloor (d_{\min} - 1)/2 \rfloor = 2^{k-2} - 1$$

errors.

Augmented Hadamard codes are most useful for getting short messages through channels with very high noise.

Summary for augmented Hadamard codes

- Possible parameters: $C(2^k, k+1)$, so $(4,3)$, $(8,4)$, $(16,5)$, $(32,6)$, $(64,7)$, $(128,8)$, $(256,9), \dots$
- Construction: start with the $k \times 2^k$ matrix whose columns are all different vectors of length k, in lexicographical order, then add an all-1 row to obtain the generator matrix G'.
- $d_{\text{min}} = 2^{k-1}$
- Can correct $2^{k-2} - 1$ errors.
- Very low code rate $(k/2^k)$.
- Useful for getting short messages through channels with very high noise.