COMP2610/6261 - Information Theory

Lecture 21: Hamming Codes & Coding Review

Robert C. Williamson

Research School of Computer Science

Australian National University

16 October, 2018

Today's plan

Noisy Channel Coding Theorem proves there exists codes with rate $R < C$ with arbitrarily low probability of error.

Today's plan

Noisy Channel Coding Theorem proves there exists codes with rate $R < C$ with arbitrarily low probability of error.
But proof was non-constructive — we used a random code in order to be able to actually calculate error probability.

Today's plan

Noisy Channel Coding Theorem proves there exists codes with rate $R < C$ with arbitrarily low probability of error.
But proof was non-constructive — we used a random code in order to be able to actually calculate error probability.
What about constructive codes?

Today's plan

Noisy Channel Coding Theorem proves there exists codes with rate $R < C$ with arbitrarily low probability of error.

But proof was non-constructive — we used a random code in order to be able to actually calculate error probability.

What about constructive codes?

We will focus on linear codes amd look at two simple linear codes:

- repetition codes
- Hamming codes

We will sketch what can be said about the rate and reliability of the latter

Repetition Codes

Simplest channel code: add *redundancy* by repeating every bit of the message (say) 3 times:

Source sequence **s**	Transmitted sequence **t**
0	0 0 0
1	1 1 1

This *repetition code* is called R_3.

Repetition Codes for the BSC

Example

On a binary symmetric channel with flip probability f, receiver sees

$$\boldsymbol{r} = \boldsymbol{t} + \boldsymbol{\eta}$$

where $\boldsymbol{\eta}$ is a *noise* vector

- $p(\eta_i = 1) = f$

Repetition Codes for the BSC

Example

On a binary symmetric channel with flip probability f, receiver sees

$$\boldsymbol{r} = \boldsymbol{t} + \boldsymbol{\eta}$$

where $\boldsymbol{\eta}$ is a *noise* vector

- $p(\eta_i = 1) = f$

Example setting of $\boldsymbol{\eta}$, and resulting message $\boldsymbol{r}$:

s	0	0	1	0	1	1	0
t	0 0 0	0 0 0	1 1 1	0 0 0	1 1 1	1 1 1	0 0 0
$\boldsymbol{\eta}$	0 0 0	0 0 1	0 0 0	0 0 0	1 0 1	0 0 0	0 0 0
r	0 0 0	0 0 1	1 1 1	0 0 0	0 1 0	1 1 1	0 0 0

Note that elements of $\boldsymbol{\eta}$ are not replicated like those of $\boldsymbol{t}$

- noise acts independently on every bit

Beyond Repetition Codes

Goal: Communication with small probability of error and high rate:

- Repetition codes introduce redundancy on a per-bit basis
- Can we improve on this?

Beyond Repetition Codes

Goal: Communication with small probability of error and high rate:

- Repetition codes introduce redundancy on a per-bit basis
- Can we improve on this?

Beyond Repetition Codes

Goal: Communication with small probability of error and high rate:

- Repetition codes introduce redundancy on a per-bit basis
- Can we improve on this?

Idea: Introduce redundancy to blocks of data instead

Beyond Repetition Codes

Goal: Communication with small probability of error and high rate:

- Repetition codes introduce redundancy on a per-bit basis
- Can we improve on this?

Idea: Introduce redundancy to blocks of data instead

Block Code

A block code is a rule for encoding a length-K sequence of source bits **s** into a length-N sequence of transmitted bits **t**.

- Introduce redundancy: $N > K$
- Focus on *Linear codes*

We will introduce a simple type of block code called
the (7,4) Hamming code

An Example

The (7, 4) Hamming Code

Consider $K = 4$, and a source message $\mathbf{s} = 1\ 0\ 0\ 0$

The repetition code R_2 produces

$$\mathbf{t} = 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0$$

The (7,4) Hamming code produces

$$\mathbf{t} = 1\ 0\ 0\ 0\ 1\ 0\ 1$$

- Redundancy, but not repetition
- How are these magic bits computed?

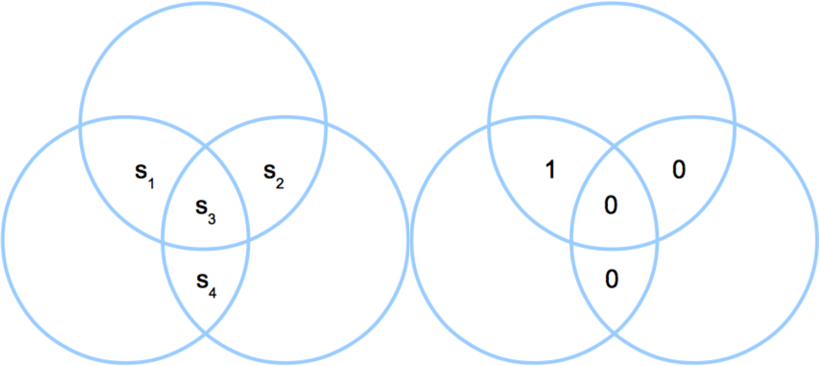
The (7,4) Hamming code
Coding

Consider $K = 4$, $N = 7$ and $\mathbf{s} = 1\ 0\ 0\ 0$

The (7,4) Hamming code

Coding

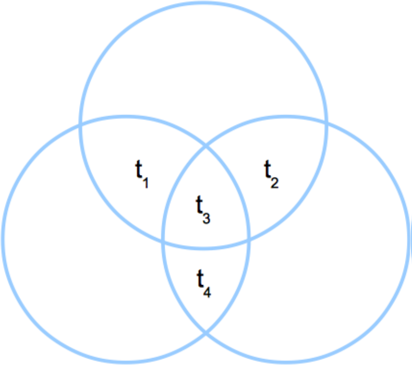
Consider $K = 4$, $N = 7$ and $\mathbf{s} = 1\ 0\ 0\ 0$

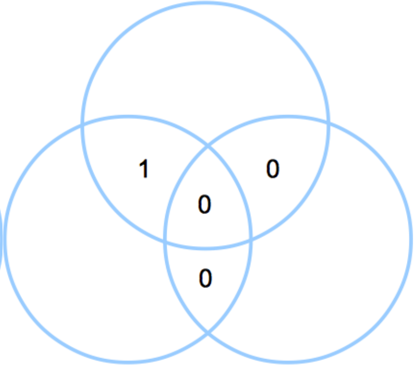


The (7,4) Hamming code

Coding

Copy the source bits into the the first 4 target bits:

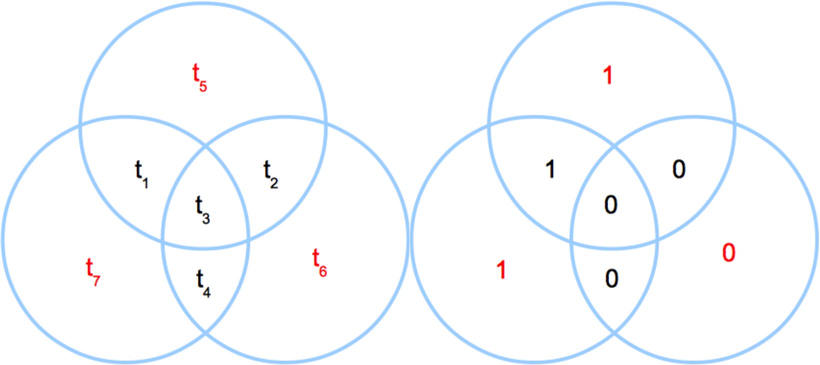




The (7,4) Hamming code

Coding

Set *parity-check* bits so that the number of ones within each circle is even:



So we have $\mathbf{s} = 1\ 0\ 0\ 0 \xrightarrow{\text{encoder}} \mathbf{t} = 1\ 0\ 0\ 0\ 1\ 0\ 1$

The (7,4) Hamming code

Algebraically, we have set:

$$
\begin{aligned}
t_i &= s_i \text{ for } i = 1, \ldots, 4 \\
t_5 &= s_1 \oplus s_2 \oplus s_3 \\
t_6 &= s_2 \oplus s_3 \oplus s_4 \\
t_7 &= s_1 \oplus s_3 \oplus s_4
\end{aligned}
$$

where we use modulo-2 arithmetic

The (7,4) Hamming code

Coding

In matrix form:

$$\mathbf{t} = \mathbf{G}^T\mathbf{s} \text{ with } \mathbf{G}^T = \begin{bmatrix} \mathbf{I}_4 \\ \mathbf{P} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix},$$

where $\mathbf{s} = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 \end{bmatrix}^T$

G is called the *Generator matrix* of the code.

The Hamming code is linear!

The (7,4) Hamming code:

Codewords

Each (unique) sequence that can be transmitted is called a *codeword*.

	s	Codeword (**t**)
	0010	0010111
Codeword examples:	0110	0110001
	1010	1010010
	1110	?

For the (7,4) Hamming code we have a total of 16 codewords

The (7,4) Hamming code

Codewords

Write

$$\mathbf{G}^T = \begin{bmatrix} \mathbf{G}_{1\cdot} & \mathbf{G}_{2\cdot} & \mathbf{G}_{3\cdot} & \mathbf{G}_{4\cdot} \end{bmatrix}$$

where each $\mathbf{G}_{i\cdot}$ is a 7 dimensional bit vector

Then, the transmitted message is

$$\begin{aligned} \mathbf{t} &= \mathbf{G}^T \mathbf{s} \\ &= \begin{bmatrix} \mathbf{G}_{1\cdot} & \mathbf{G}_{2\cdot} & \mathbf{G}_{3\cdot} & \mathbf{G}_{4\cdot} \end{bmatrix} \mathbf{s} \\ &= s_1 \mathbf{G}_{1\cdot} + \ldots + s_4 \mathbf{G}_{4\cdot} \end{aligned}$$

The (7,4) Hamming code:

Codewords

There are 2^7 possible transmitted bit strings

- There are $2^7 - 2^4$ other bit strings that immediately imply corruption

- If we see a codeword, does that imply no corruption?

Any two codewords differ in at least three bits

- Each original bit belongs to at least two circles

- Useful in constructing reliable decoders

The (7,4) Hamming code:

Decoding

We can encode a length-4 sequence **s** into a length-7 sequence **t** using 3 parity check bits

The (7,4) Hamming code:
Decoding

We can encode a length-4 sequence **s** into a length-7 sequence **t** using 3 parity check bits

t can be corrupted by noise which can flip *any* of the 7 bits (including the parity check bits):

$$
\begin{array}{cc}
\mathbf{s} & 1\ 0\ 0\ 0 \\
\mathbf{t} & \overbrace{1\ 0\ 0\ 0\ 1\ 0\ 1} \\
\eta & 0\ 1\ 0\ 0\ 0\ 0\ 0 \\
\hline
\mathbf{r} & 1\ 1\ 0\ 0\ 1\ 0\ 1
\end{array}
$$

The (7,4) Hamming code:

Decoding

We can encode a length-4 sequence **s** into a length-7 sequence **t** using 3 parity check bits

t can be corrupted by noise which can flip *any* of the 7 bits (including the parity check bits):

$$
\begin{array}{cc}
\mathbf{s} & 1\,0\,0\,0 \\
\mathbf{t} & \overbrace{1\,0\,0\,0\,1\,0\,1} \\
\eta & 0\,1\,0\,0\,0\,0\,0 \\
\hline
\mathbf{r} & 1\,1\,0\,0\,1\,0\,1
\end{array}
$$

How should we decode **r**?

- We could do this exhaustively using the 16 codewords
- Assuming BSC, uniform $p(\mathbf{s})$: Get the most probable explanation
- Find **s** such that $\|\mathbf{t}(\mathbf{s}) \ominus \mathbf{r}\|_1$ is minimum

The (7,4) Hamming code:

Decoding

We can encode a length-4 sequence **s** into a length-7 sequence **t** using 3 parity check bits

t can be corrupted by noise which can flip *any* of the 7 bits (including the parity check bits):

$$
\begin{array}{cl}
\mathbf{s} & 1\,0\,0\,0 \\
\mathbf{t} & \overbrace{1\,0\,0\,0\,1\,0\,1} \\
\eta & 0\,1\,0\,0\,0\,0\,0 \\
\hline
\mathbf{r} & 1\,1\,0\,0\,1\,0\,1
\end{array}
$$

How should we decode **r**?

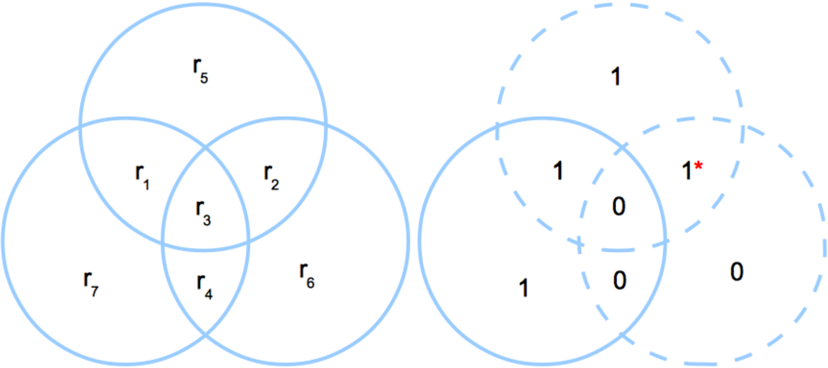
- We could do this exhaustively using the 16 codewords
- Assuming BSC, uniform $p(\mathbf{s})$: Get the most probable explanation
- Find **s** such that $\|\mathbf{t}(\mathbf{s}) \ominus \mathbf{r}\|_1$ is minimum

We can get the most probable source vector in an more *efficient* way.

The (7,4) Hamming code:

Decoding Example 1

We have $\mathbf{s} = 1\ 0\ 0\ 0 \xrightarrow{\text{encoder}} \mathbf{t} = 1\ 0\ 0\ 0\ 1\ 0\ 1 \xrightarrow{\text{noise}} \mathbf{r} = 1\ 1\ 0\ 0\ 1\ 0\ 1$:



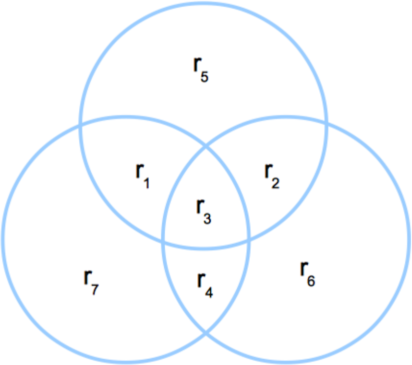
(1) Detect circles with wrong (odd) parity

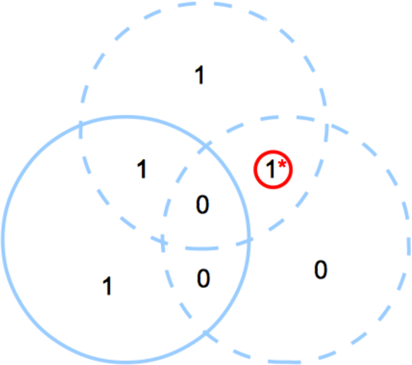
- What bit is responsible for this?

The (7,4) Hamming code:

Decoding Example 1

We have $\mathbf{s} = 1\ 0\ 0\ 0 \xrightarrow{\text{encoder}} \mathbf{t} = 1\ 0\ 0\ 0\ 1\ 0\ 1 \xrightarrow{\text{noise}} \mathbf{r} = 1\ 1\ 0\ 0\ 1\ 0\ 1$:





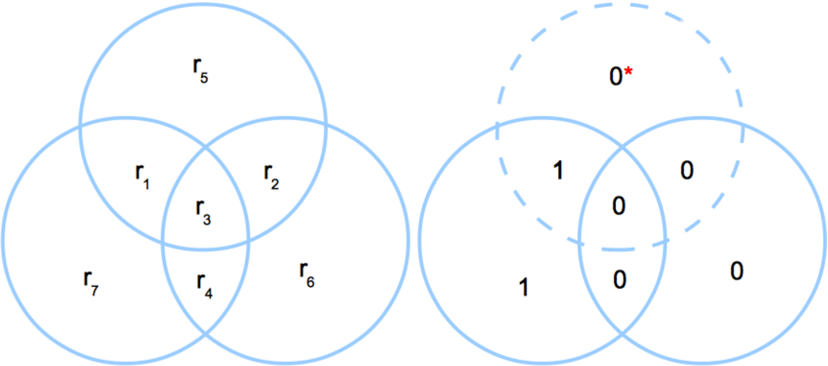
(2) Detect culprit bit and flip it

- The decoded sequence is $\hat{\mathbf{s}} = 1\ 0\ 0\ 0$

The (7,4) Hamming code:

Decoding Example 2

We have $\mathbf{s} = 1\ 0\ 0\ 0 \xrightarrow{\text{encoder}} \mathbf{t} = 1\ 0\ 0\ 0\ 1\ 0\ 1 \xrightarrow{\text{noise}} \mathbf{r} = 1\ 0\ 0\ 0\ 0\ 0\ 1$:



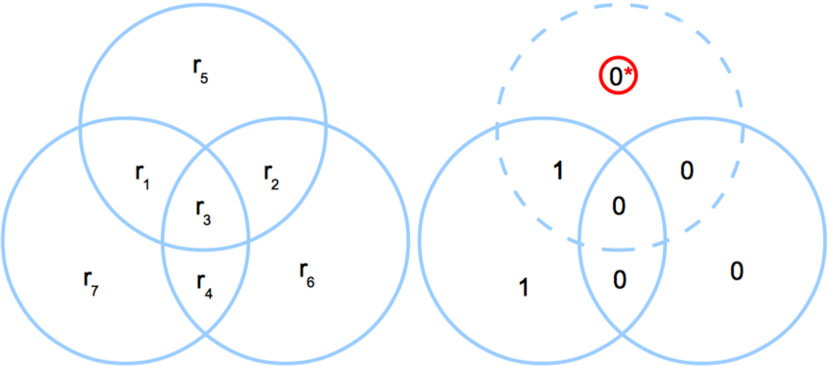
(1) Detect circles with wrong (odd) parity

- What bit is responsible for this?

The (7,4) Hamming code:

Decoding Example 2

We have $\mathbf{s} = 1\ 0\ 0\ 0 \xrightarrow{\text{encoder}} \mathbf{t} = 1\ 0\ 0\ 0\ 1\ 0\ 1 \xrightarrow{\text{noise}} \mathbf{r} = 1\ 0\ 0\ 0\ 0\ 0\ 1$:



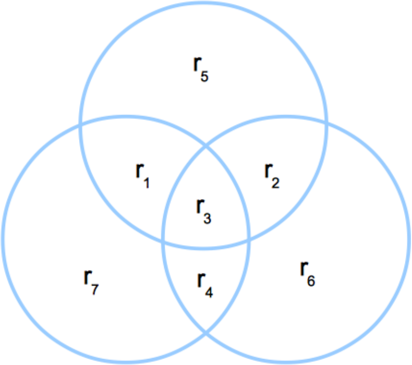
(2) Detect culprit bit and flip it

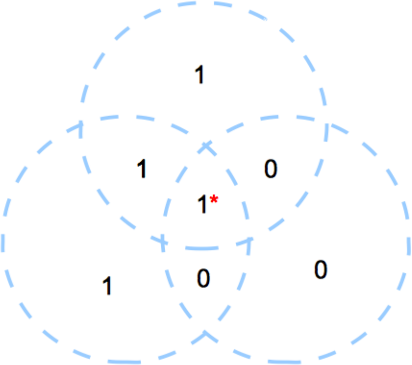
- The decoded sequence is $\hat{\mathbf{s}} = 1\ 0\ 0\ 0$

The (7,4) Hamming code:

Decoding Example 3

We have $\mathbf{s} = 1\ 0\ 0\ 0 \xrightarrow{\text{encoder}} \mathbf{t} = 1\ 0\ 0\ 0\ 1\ 0\ 1 \xrightarrow{\text{noise}} \mathbf{r} = 1\ 0\ 1\ 0\ 1\ 0\ 1$:





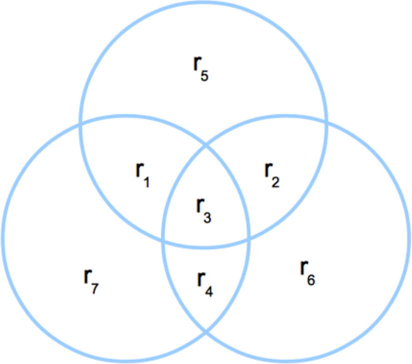
(1) Detect circles with wrong (odd) parity

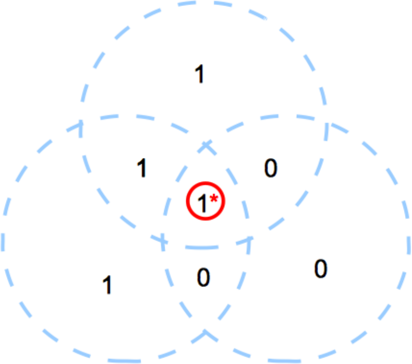
- What bit is responsible for this?

The (7,4) Hamming code:

Decoding Example 3

We have $\mathbf{s} = 1\ 0\ 0\ 0 \xrightarrow{\text{encoder}} \mathbf{t} = 1\ 0\ 0\ 0\ 1\ 0\ 1 \xrightarrow{\text{noise}} \mathbf{r} = 1\ 0\ 1\ 0\ 1\ 0\ 1$:





(2) Detect culprit bit and flip it

- The decoded sequence is $\hat{\mathbf{s}} = 1\ 0\ 0\ 0$

The (7,4) Hamming code:

Optimal Decoding Algorithm: Syndrome Decoding

Given $\mathbf{r} = r_1, \ldots, r_7$, assume BSC with small noise level f:

1. Define the syndrome as the length-3 vector $\mathbf{z}$ that describes the pattern of violations of the parity bits r_5, r_6, r_7.
    - $\mathbf{z}_i = 1$ when the ith parity bit does not match the parity of $\mathbf{r}$
    - Flipping a single bit leads to a different syndrome

The (7,4) Hamming code:

Optimal Decoding Algorithm: Syndrome Decoding

Given $\mathbf{r} = r_1, \ldots, r_7$, assume BSC with small noise level f:

1. Define the syndrome as the length-3 vector $\mathbf{z}$ that describes the pattern of violations of the parity bits r_5, r_6, r_7.
    - $\mathbf{z}_i = 1$ when the ith parity bit does not match the parity of $\mathbf{r}$
    - Flipping a single bit leads to a different syndrome
2. Check parity bits r_5, r_6, r_7 and identify the syndrome

The (7,4) Hamming code:

Optimal Decoding Algorithm: Syndrome Decoding

Given $\mathbf{r} = r_1, \ldots, r_7$, assume BSC with small noise level f:

1. Define the syndrome as the length-3 vector $\mathbf{z}$ that describes the pattern of violations of the parity bits r_5, r_6, r_7.
   - $\mathbf{z}_i = 1$ when the ith parity bit does not match the parity of $\mathbf{r}$
   - Flipping a single bit leads to a different syndrome
2. Check parity bits r_5, r_6, r_7 and identify the syndrome
3. Unflip the *single bit* responsible for this pattern of violation
   - This syndrome could have been caused by other noise patterns

The (7,4) Hamming code:

Optimal Decoding Algorithm: Syndrome Decoding

Given $\mathbf{r} = r_1, \ldots, r_7$, assume BSC with small noise level f:

1. Define the syndrome as the length-3 vector $\mathbf{z}$ that describes the pattern of violations of the parity bits r_5, r_6, r_7.
   - $\mathbf{z}_i = 1$ when the ith parity bit does not match the parity of $\mathbf{r}$
   - Flipping a single bit leads to a different syndrome
2. Check parity bits r_5, r_6, r_7 and identify the syndrome
3. Unflip the *single bit* responsible for this pattern of violation
   - This syndrome could have been caused by other noise patterns

$\mathbf{z}$	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
Flip bit	none	r_7	r_6	r_4	r_5	r_1	r_2	r_3

The (7,4) Hamming code:

Optimal Decoding Algorithm: Syndrome Decoding

Given $\mathbf{r} = r_1, \ldots, r_7$, assume BSC with small noise level f:

1. Define the syndrome as the length-3 vector $\mathbf{z}$ that describes the pattern of violations of the parity bits r_5, r_6, r_7.
    - $\mathbf{z}_i = 1$ when the ith parity bit does not match the parity of $\mathbf{r}$
    - Flipping a single bit leads to a different syndrome
2. Check parity bits r_5, r_6, r_7 and identify the syndrome
3. Unflip the *single bit* responsible for this pattern of violation
    - This syndrome could have been caused by other noise patterns

$\mathbf{z}$	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
Flip bit	none	r_7	r_6	r_4	r_5	r_1	r_2	r_3

The optimal decoding algorithm unflips at most one bit

The (7,4) Hamming code:

Syndrome Decoding: Matrix Form

Recall that we just need to compare the expected parity bits with the actual ones we received:

$$
\begin{aligned}
z_1 &= r_1 \oplus r_2 \oplus r_3 \ominus r_5 \\
z_2 &= r_2 \oplus r_3 \oplus r_4 \ominus r_6 \\
z_3 &= r_1 \oplus r_3 \oplus r_4 \ominus r_7,
\end{aligned}
$$

The (7,4) Hamming code:
Syndrome Decoding: Matrix Form

Recall that we just need to compare the expected parity bits with the actual ones we received:

$$
\begin{aligned}
z_1 &= r_1 \oplus r_2 \oplus r_3 \ominus r_5 \\
z_2 &= r_2 \oplus r_3 \oplus r_4 \ominus r_6 \\
z_3 &= r_1 \oplus r_3 \oplus r_4 \ominus r_7,
\end{aligned}
$$

but in modulo-2 arithmetic $-1 \equiv 1$ so we can replace $\ominus$ with $\oplus$ so we have:

$$
\mathbf{z} = \mathbf{Hr} \text{ with } \mathbf{H} = \begin{bmatrix} \mathbf{P} & \mathbf{I}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}
$$

The (7,4) Hamming code:

Syndrome Decoding: Matrix Form

Recall that we just need to compare the expected parity bits with the actual ones we received:

$$z_1 = r_1 \oplus r_2 \oplus r_3 \ominus r_5$$
$$z_2 = r_2 \oplus r_3 \oplus r_4 \ominus r_6$$
$$z_3 = r_1 \oplus r_3 \oplus r_4 \ominus r_7,$$

but in modulo-2 arithmetic $-1 \equiv 1$ so we can replace $\ominus$ with $\oplus$ so we have:

$$\mathbf{z} = \mathbf{H}\mathbf{r} \text{ with } \mathbf{H} = \begin{bmatrix} \mathbf{P} & \mathbf{I}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Homework: What is the syndrome for a codeword?

The (7,4) Hamming code:

Optimal Decoding Algorithm: Syndrome Decoding

When the noise level f on the BSC is small, it may be reasonable that we see only a single bit flip in a sequence of 4 bits

The syndrome decoding method exactly recovers the source message in this case

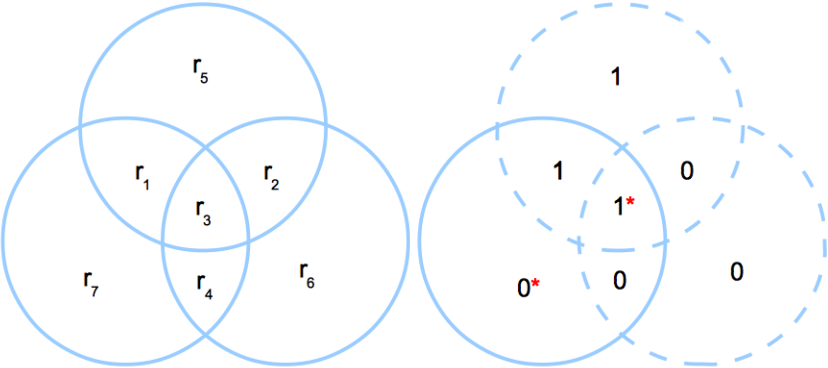
- c.f. Noise flipping one bit in the repetition code R_3

But what happens if the noise flips more than one bit?

The (7,4) Hamming code:

Decoding Example 4: Flipping 2 Bits

We have $\mathbf{s} = 1\ 0\ 0\ 0 \xrightarrow{\text{encoder}} \mathbf{t} = 1\ 0\ 0\ 0\ 1\ 0\ 1 \xrightarrow{\text{noise}} \mathbf{r} = 1\ 0\ 1\ 0\ 1\ 0\ 0$:



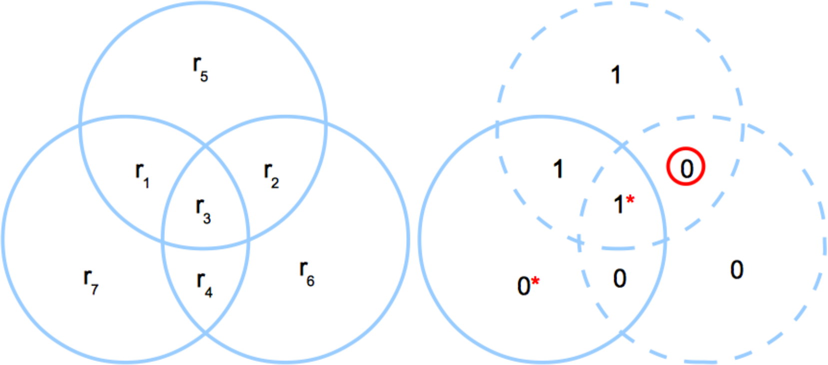
(1) Detect circles with wrong (odd) parity

- What bit is responsible for this?

The (7,4) Hamming code:

Decoding Example 4: Flipping 2 Bits

We have $\mathbf{s} = 1\ 0\ 0\ 0 \xrightarrow{\text{encoder}} \mathbf{t} = 1\ 0\ 0\ 0\ 1\ 0\ 1 \xrightarrow{\text{noise}} \mathbf{r} = 1\ 0\ 1\ 0\ 1\ 0\ 0$:



(2) Detect culprit bit and flip it

- The decoded sequence is $\hat{\mathbf{s}} = 1\ 1\ 1\ 0$
  - We have made 3 errors but only 2 involve the actual message

The (7,4) Hamming code:

Decoding Exercises

[Mackay, Ex 1.5]: Decode the following sequences using the syndrome decoding for the (7,4) Hamming code:

(a) $\mathbf{r} = 1101011 \rightarrow \hat{\mathbf{s}} =??$

(b) $\mathbf{r} = 0110110 \rightarrow \hat{\mathbf{s}} =??$

(c) $\mathbf{r} = 0100111 \rightarrow \hat{\mathbf{s}} =??$

(d) $\mathbf{r} = 1111111 \rightarrow \hat{\mathbf{s}} =??$

Work out the answers on your own.

The (7,4) Hamming code: Solution

For each exercise we simply compute the *syndrome* and use the optimal decoding algorithm (Table above) to determine which bit we should unflip.

(a) $\mathbf{r} = 1101011 \rightarrow$: $z_1 = r_1 \oplus r_2 \oplus r_3 \oplus r_5 = 0 \quad z_2 = r_2 \oplus r_3 \oplus r_4 \oplus r_6 = 1 \quad z_3 = r_1 \oplus r_3 \oplus r_4 \oplus r_7 = 1$ Therefore $\mathbf{z} = 011$, we unflip r_4, $\hat{\mathbf{s}} = 1100$

(b) $\mathbf{r} = 0110110 \rightarrow \mathbf{z} = 111$, we unflip r_3, $\hat{\mathbf{s}} = 0100$

(c) $\mathbf{r} = 0100111 \rightarrow \mathbf{z} = 001$, we unflip r_7, $\hat{\mathbf{s}} = 0100$

(d) $\mathbf{r} = 1111111 \rightarrow \mathbf{z} = 000$, we don't unflip any bit, $\hat{\mathbf{s}} = 1111$

The (7,4) Hamming code:

Zero-Syndrome Noise Vectors

[Mackay, Ex 1.7] Find some noise vectors that give the all-zero syndrome (so that the optimal decoding algorithm will not correct them). How many of these vectors are there?

Solution

By definition we have that the all-zero syndrome implies that the corresponding noise components should cancel out. For example for the first component we have:
$z_1 = r_1 \oplus r_2 \oplus r_3 \oplus r_5 = t_1 \oplus t_2 \oplus t_3 \oplus t_5 \oplus \eta_1 \oplus \eta_2 \oplus \eta_3 \oplus \eta_5$. But $t_i = s_i$ for $i = 1, \ldots, 4$ and $t_5 = s_1 \oplus s_2 \oplus s_3$. Therefore
$z_1 = 2s_1 \oplus 2s_2 \oplus 2s_3 \oplus \eta_1 \oplus \eta_2 \oplus \eta_3 \oplus \eta_5 = \eta_1 \oplus \eta_2 \oplus \eta_3 \oplus \eta_5$. Thus, we have:

$$z_1 = \eta_1 \oplus \eta_2 \oplus \eta_3 \oplus \eta_5 = 0$$
$$z_2 = \eta_2 \oplus \eta_3 \oplus \eta_4 \oplus \eta_6 = 0$$
$$z_3 = \eta_1 \oplus \eta_3 \oplus \eta_4 \oplus \eta_7 = 0$$

which is equivalent to:

$$\eta_5 = \eta_1 \oplus \eta_2 \oplus \eta_3$$
$$\eta_6 = \eta_2 \oplus \eta_3 \oplus \eta_4$$
$$\eta_7 = \eta_1 \oplus \eta_3 \oplus \eta_4$$

Solution (cont.)

As η_5 is determined by η_1, η_2, η_3 we have $2^3 = 8$ possibilities here.
Now, for fixed η_1, η_2 (and η_3) in the previous step we only have two possibilities for η_4, which determines η_6.
We have now that all the variables are set and η_7 is fully determined by their values.
Thus, we have $8 \times 2 \times 1$ possible noise vectors that yield the all-zero syndrome.
The trivial noise vectors that yield this syndrome are: $\boldsymbol{\eta} = 0000000$ and $\boldsymbol{\eta} = 1111111$.
However, we can follow the above procedure and set the corresponding variables.
This is equivalent to having arbitrary settings for η_1, η_2, η_3 and η_4 which gives us 16 possible noise vectors which exactly correspond to the 16 codewords of the (7,4) Hamming code.

The (7,4) Hamming code:

Error Probabilities

Decoding Error : Occurs if at least one of the decoded bits $\hat{s}_i$ does not match the corresponding source bit s_i for $i = 1, \ldots 4$

The (7,4) Hamming code:

Error Probabilities

Decoding Error : Occurs if at least one of the decoded bits $\hat{s}_i$ does not match the corresponding source bit s_i for $i = 1, \ldots 4$

p(Block Error) : $p_B = p(\hat{\mathbf{s}} \neq \mathbf{s})$

The (7,4) Hamming code:

Error Probabilities

Decoding Error : Occurs if at least one of the decoded bits $\hat{s}_i$ does not match the corresponding source bit s_i for $i = 1, \ldots 4$

p(Block Error) : $p_B = p(\hat{\mathbf{s}} \neq \mathbf{s})$

p(Bit Error) : $p_b = \frac{1}{K} \sum_{k=1}^{K} p(\hat{s}_k \neq s_k)$

The (7,4) Hamming code:

Error Probabilities

Decoding Error : Occurs if at least one of the decoded bits $\hat{s}_i$ does not match the corresponding source bit s_i for $i = 1, \ldots 4$

p(Block Error) : $p_B = p(\hat{\mathbf{s}} \neq \mathbf{s})$

p(Bit Error) : $p_b = \frac{1}{K} \sum_{k=1}^{K} p(\hat{s}_k \neq s_k)$

Rate : $R = \frac{K}{N} = \frac{4}{7}$

The (7,4) Hamming code:

Error Probabilities

Decoding Error : Occurs if at least one of the decoded bits $\hat{s}_i$ does not match the corresponding source bit s_i for $i = 1, \ldots 4$

p(Block Error) : $p_B = p(\hat{\mathbf{s}} \neq \mathbf{s})$

p(Bit Error) : $p_b = \frac{1}{K} \sum_{k=1}^{K} p(\hat{s}_k \neq s_k)$

Rate : $R = \frac{K}{N} = \frac{4}{7}$

What is the probability of block error for the (7,4) Hamming code with $f = 0.1$?

The (7,4) Hamming code:

Leading-Term Error Probabilities

Block Error: This occurs when 2 or more bits in the block of 7 are flipped

We can approximate p_B to the leading term:

$$p_B = \sum_{m=2}^{7} \binom{7}{m} f^m (1-f)^{7-m}$$
$$\approx \binom{7}{2} f^2 = 21 f^2.$$

The (7,4) Hamming code:

Leading-Term Error Probabilities

Bit Error: Given that a block error occurs, the noise must corrupt 2 or more bits

The most probable case is when the noise corrupts 2 bits, which induces 3 errors in the decoded vector:

The (7,4) Hamming code:

Leading-Term Error Probabilities

Bit Error: Given that a block error occurs, the noise must corrupt 2 or more bits

The most probable case is when the noise corrupts 2 bits, which induces 3 errors in the decoded vector:

- $p(\hat{s}_i \neq s_i) \approx \frac{3}{7} p_B$ for $i = 1, \ldots, 7$

The (7,4) Hamming code:

Leading-Term Error Probabilities

Bit Error: Given that a block error occurs, the noise must corrupt 2 or more bits

The most probable case is when the noise corrupts 2 bits, which induces 3 errors in the decoded vector:

- $p(\hat{s}_i \neq s_i) \approx \frac{3}{7} p_B$ for $i = 1, \ldots, 7$
- All bits are equally likely to be corrupted (due to symmetry)

The (7,4) Hamming code:

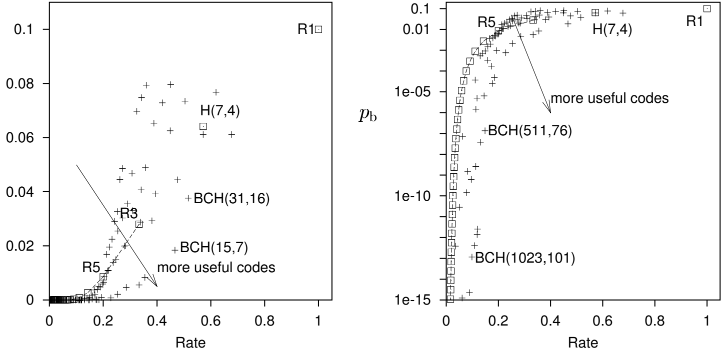
Leading-Term Error Probabilities

Bit Error: Given that a block error occurs, the noise must corrupt 2 or more bits

The most probable case is when the noise corrupts 2 bits, which induces 3 errors in the decoded vector:

- $p(\hat{s}_i \neq s_i) \approx \frac{3}{7} p_B$ for $i = 1, \ldots, 7$
- All bits are equally likely to be corrupted (due to symmetry)
- $p_b \approx \frac{3}{7} p_B \approx 9f^2$

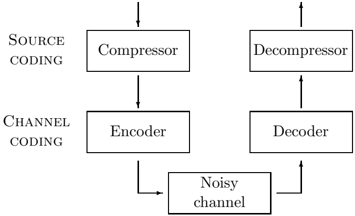
What Can Be Achieved with Hamming Codes?



- H(7,4) improves p_b at a moderate rate $R = 4/7$
- BCH are a generalization of Hamming codes.

Coding: Review

The Big Picture



Source Coding for Compression

- Shrink sequences
- Identify and remove redundancy
- Size limited by entropy
- Source Coding Theorems (Block & Variable Length)

Channel Coding for Reliability

- Protect sequences
- Add known form of redundancy
- Rate limited by capacity
- Noisy-Channel Coding Theorem