COMP2610/6261 - Information Theory

Lecture 22: Hamming Codes

Mark Reid and **Aditya Menon**

Research School of Computer Science
The Australian National University

Australian
National
University

October 15th, 2014

Reminder: Repetition Codes

The repetition code R_3:

$$
\begin{array}{c|ccccccc}
\mathbf{s} & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
\mathbf{t} & \overbrace{0\,0\,0} & \overbrace{0\,0\,0} & \overbrace{1\,1\,1} & \overbrace{0\,0\,0} & \overbrace{1\,1\,1} & \overbrace{1\,1\,1} & \overbrace{0\,0\,0} \\
\boldsymbol{\eta} & 0\,0\,0 & 0\,0\,1 & 0\,0\,0 & 0\,0\,0 & 1\,0\,1 & 0\,0\,0 & 0\,0\,0 \\
\hline
\mathbf{r} & 0\,0\,0 & 0\,0\,1 & 1\,1\,1 & 0\,0\,0 & 0\,1\,0 & 1\,1\,1 & 0\,0\,0
\end{array}
$$

For a BSC with bit flip probability $f = 0.1$, drives error rate down to $\approx 3\%$

For general f, the error probability is $f^2(3 - 2f)$

This time

- Introduction to block codes
  - Extension to basic repetition codes
- The (7,4) Hamming code
- Redundancy in (linear) block codes through parity check bits
- Syndrome decoding

Motivation

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

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

Motivation

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

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

Motivation

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

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

Idea: Introduce redundancy to blocks of data instead

Motivation

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

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

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 $\mathbf{s}$ into a length-N sequence of transmitted bits $\mathbf{t}$.

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

Motivation

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

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

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

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

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

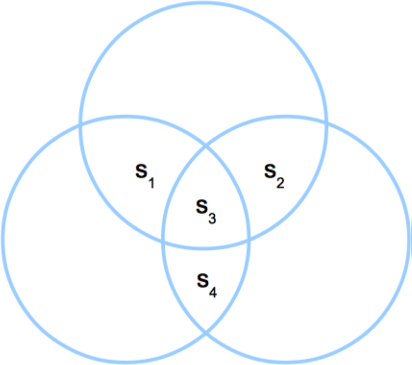
- Redundancy, but not repetition
- How are these magic bits computed?

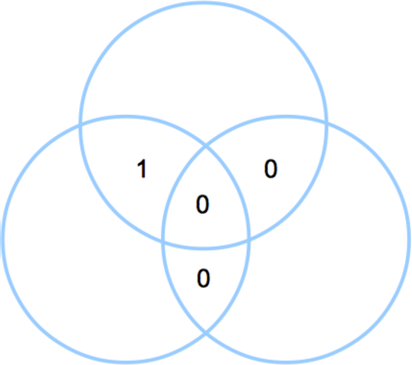
Consider $K = 4$, $N = 7$ and $\mathbf{s} = 1\ 0\ 0\ 0$

It will help to think of the code in terms of overlapping circles

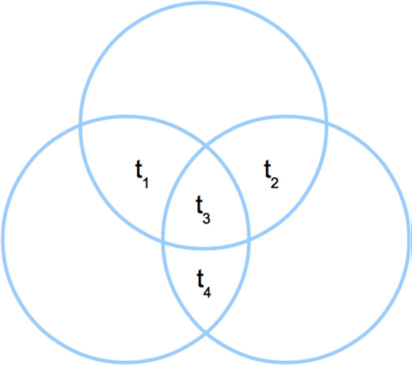
Consider $K = 4$, $N = 7$ and $\mathbf{s} = 1\ 0\ 0\ 0$

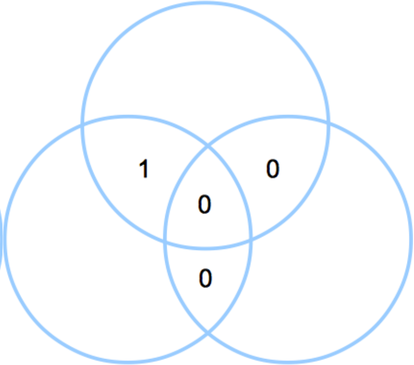
It will help to think of the code in terms of overlapping circles



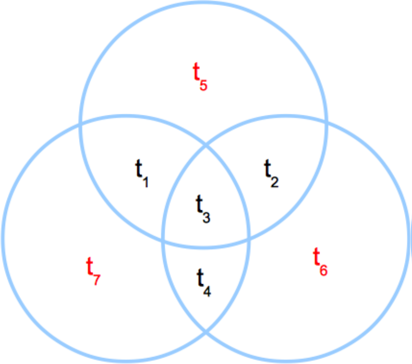


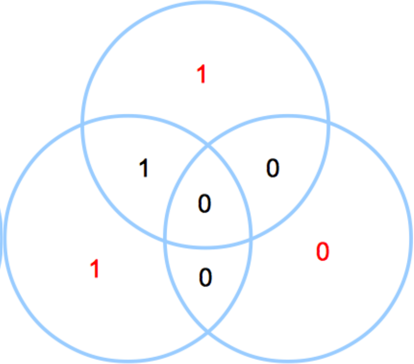
Copy the source bits into the the first 4 target bits:





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
Coding

It is clear that 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

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*.

	$\mathbf{s}$	Codeword ($\mathbf{t}$)
	0010	0010111
Codeword examples:	0110	0110001
	1010	1010010
	1110	?

The (7,4) Hamming code:
Codewords

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

	$\mathbf{s}$	Codeword ($\mathbf{t}$)
	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

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

	$\mathbf{s}$	Codeword ($\mathbf{t}$)
	0010	0010111
Codeword examples:	0110	0110001
	1010	1010010
	1110	?

- For the (7,4) Hamming code we have a total of 16 codewords
- There are $2^7 - 2^4$ other bit strings that immediately imply corruption

The (7,4) Hamming code:
Codewords

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

	$\mathbf{s}$	Codeword ($\mathbf{t}$)
	0010	0010111
Codeword examples:	0110	0110001
	1010	1010010
	1110	?

- For the (7,4) Hamming code we have a total of 16 codewords
- There are $2^7 - 2^4$ other bit strings that immediately imply corruption
- Any two codewords differ in at least three bits
  - Each original bit belongs to at least two circles

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

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

All codewords can be obtained as linear combinations of the rows of $\mathbf{G}$:

$$\text{Codewords } = \left\{ \sum_{i=1}^{4} \alpha_i \mathbf{G}_{i\cdot} \right\},$$

where $\alpha_i \in \{0, 1\}$ and $\mathbf{G}_{i\cdot}$ is the $i\text{th}$ row of $\mathbf{G}$.

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

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):

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

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

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):

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

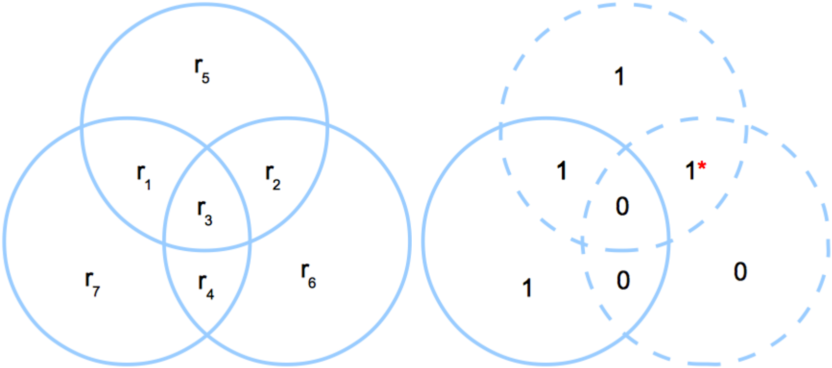
How should we decode **r**?

- 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

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

The (7,4) Hamming code:
Decoding Example 1

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

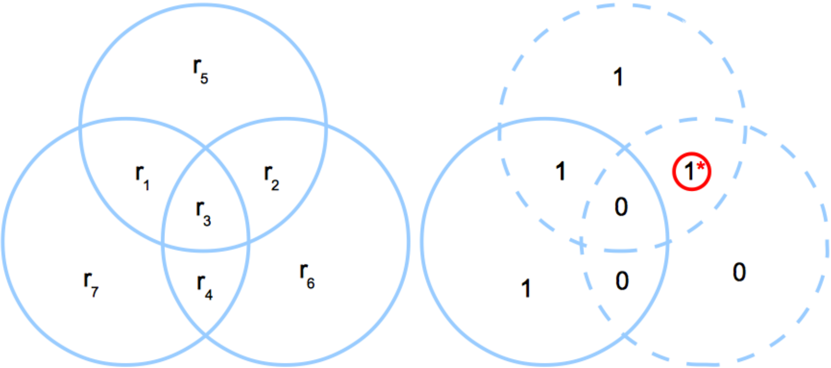


(1) Detect circles with wrong (odd) parity

- What bit is responsible for this?

The (7,4) Hamming code:
Decoding Example 1

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



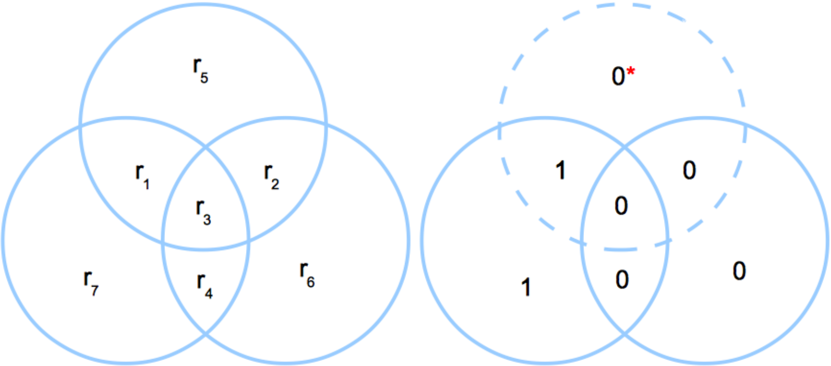
(2) Detect culprit bit and flip it

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

The (7,4) Hamming code:

Decoding Example 2

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

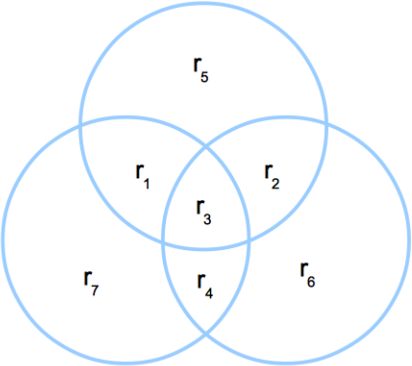


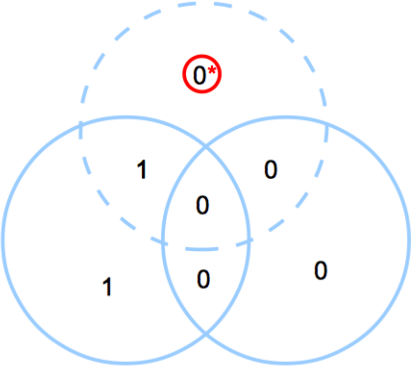
(1) Detect circles with wrong (odd) parity

- What bit is responsible for this?

The (7,4) Hamming code:
Decoding Example 2

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



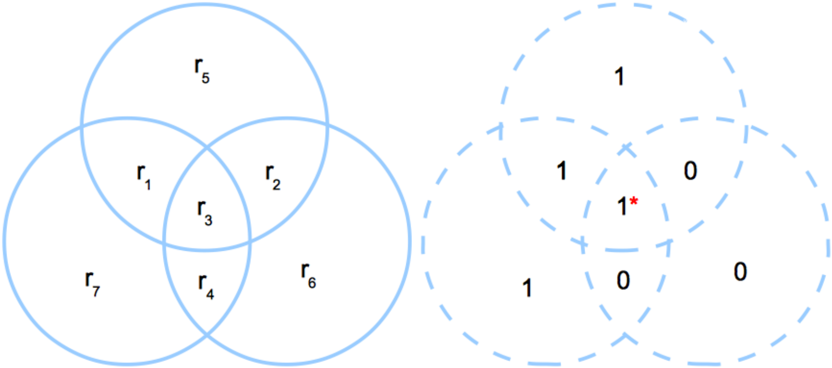


(2) Detect culprit bit and flip it

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

The (7,4) Hamming code:
Decoding Example 3

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



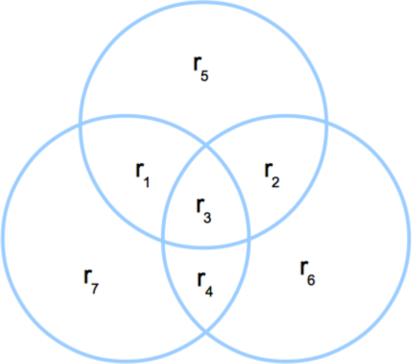
(1) Detect circles with wrong (odd) parity

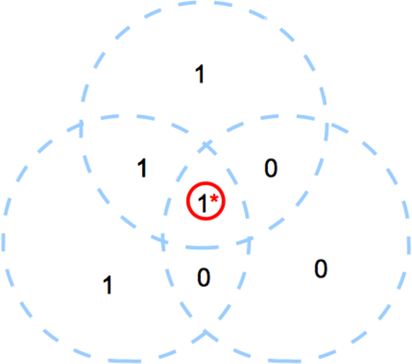
- What bit is responsible for this?

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 $\hat{\mathbf{s}} = 1\ 0\ 0\ 0$

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:

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

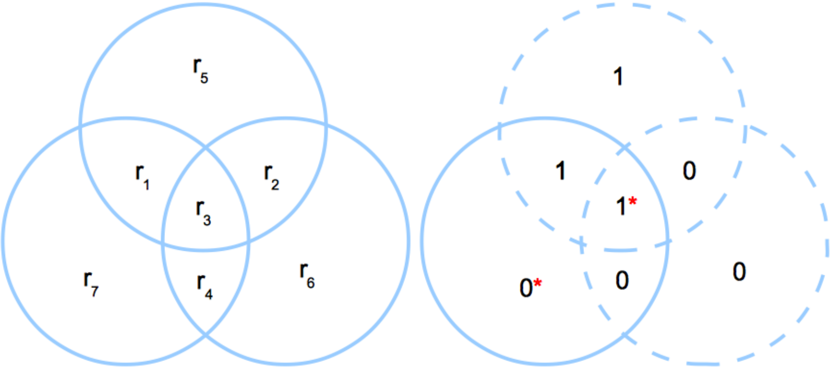
- c.f. Noise flipping one bit in the repetition code R_3

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

The (7,4) Hamming code:

Decoding Example 4: Flipping 2 Bits

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



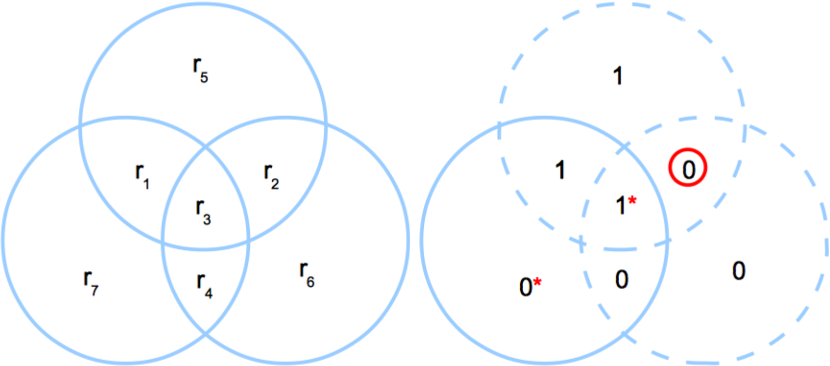
(1) Detect circles with wrong (odd) parity

- 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:
Syndrome Decoding: Matrix Form

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

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

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:

$$\mathbf{z} = \mathbf{Hr} \text{ with } \mathbf{H} = \left[\begin{array}{cc} \mathbf{P} & \mathbf{I}_3 \end{array}\right] = \left[\begin{array}{ccccccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

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:

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

What is the syndrome for a codeword?

The (7,4) Hamming code:

Syndrome Decoding: Matrix Form

Recall that we obtain a codeword with $\mathbf{t} = \mathbf{G}^T \mathbf{s}$

Assume we receive $\mathbf{r} = \mathbf{t} + \boldsymbol{\eta}$, where $\boldsymbol{\eta} = \mathbf{0}$

The syndrome is

$$\begin{aligned} \mathbf{z} &= \mathbf{H}\mathbf{r} \\ &= \mathbf{H}\mathbf{t} \\ &= \mathbf{H}\mathbf{G}^T \mathbf{s} \\ &= \mathbf{0} \end{aligned}$$

This is because $\mathbf{H}\mathbf{G}^T = \mathbf{P} + \mathbf{P} = \mathbf{0}$

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

For the noisy case we have:

$$\begin{aligned}\mathbf{r} &= \mathbf{G}^T\mathbf{s} + \boldsymbol{\eta} \\ \mathbf{z} &= \mathbf{H}\mathbf{r} \\ &= \mathbf{H}\mathbf{G}^T\mathbf{s} + \mathbf{H}\boldsymbol{\eta} \\ &= \mathbf{H}\boldsymbol{\eta}.\end{aligned}$$

For the noisy case we have:

$$\begin{aligned}
\mathbf{r} &= \mathbf{G}^T\mathbf{s} + \boldsymbol{\eta} \\
\mathbf{z} &= \mathbf{H}\mathbf{r} \\
&= \mathbf{H}\mathbf{G}^T\mathbf{s} + \mathbf{H}\boldsymbol{\eta} \\
&= \mathbf{H}\boldsymbol{\eta}.
\end{aligned}$$

Therefore, syndrome decoding boils down to find the most probable $\boldsymbol{\eta}$ satisfying $\mathbf{H}\boldsymbol{\eta} = \mathbf{z}$.

- Maximum likelihood decoder

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

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 $i = 1, \dots 4$

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:

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

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 $i = 1, \ldots, 7$

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)

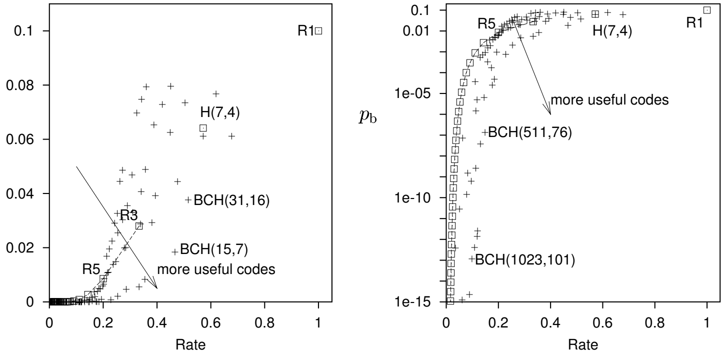
The (7,4) Hamming code:
Leading-Term Error Probabilities

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.
- BCH better than R_N but still pretty depressing

Can we do better? What is achievable / nonachievable?

Summary

- The (7,4) Hamming code
- Redundancy in (linear) block codes through parity check bits
- Syndrome decoding via identification of *single* bit noise patterns
- Block error, bit error, rate
- Reading: Mackay §1.2 – §1.5