Hamming It Up with Hamming Codes

CSE 461
Section Week 3

Error Detection/Correction

- We want to know when there are errors in communication
- Correcting errors would be even better! Why?
  - It’d save lots of time
- What are some ways we can correct errors?
  - Send data multiple times
  - Send longer symbols
  - Send data with the payload that’s a function of the payload
    - E.g., parity bits

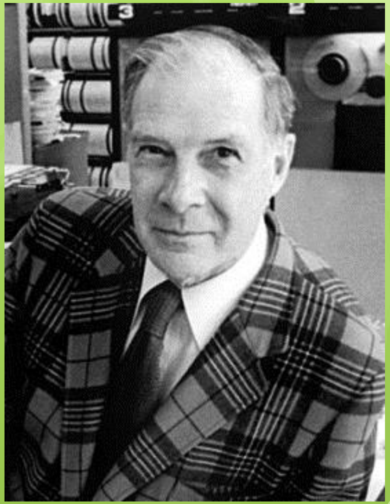
Parity Bits

- Problem
  - We want to send 1101
  - Last bit gets flipped
  - 1100 is sent instead
- How can we detect this?
  - Add another bit at the end: the sum (without carry) of all the bits
  - So instead of 1101 we send…
    - 1101**1**
  - If 1100**1** is received, we know that it's wrong—how?
    - The parity bit for 1100 should be 0, but it's not… something was flipped!
  - Even and odd parity

Error Detection/Correction

- Parity bits don’t let us correct errors (by themselves)
- Can we do any better?
- What’s the best way to detect and correct?

Hamming Codes: Background



- Richard Hamming
- Worked at Bell Labs
- Developed Hamming Codes to save time on punchcard reading errors
- Mixed message bits and parity bits to detect and correct specific errors
- Hamming codes now used for network communications as well as hard drive RAIDs

Hamming Codes: How They Work

All other bits are message bits

0 0 1 0 0 0 1 0 1 1 0 0

Bits in 1, 2, 4, 8, etc. positions are parity bits

Hamming Codes: How They Work

- We want to send 1011100
- We put its message bits into the non-2^n places, like so:

?	?	1	?	0	1	1	?	1	0	0
P_1	p_2	m_3	p_4	m_5	m_6	m_7	p_8	m_9	m_{10}	m_{11}

Hamming Codes: How They Work

- Each message bit is added to the parity bits that sum up to that message bit's place
- For m_3, 3 = 2 + 1, so we add to p_1 and p_2

+1	+1									
?	?	1	?	0	1	1	?	1	0	0
p_1	p_2	m_3	p_4	m_5	m_6	m_7	p_8	m_9	m_{10}	m_{11}

Hamming Codes: How They Work

- For m_5, 5 = 4 + 1, so we add to p_4 and p_1
  - But m_5 = 0, so we don't add anything

+1	+1									
?	?	1	?	0	1	1	?	1	0	0
p_1	p_2	m_3	p_4	m_5	m_6	m_7	p_8	m_9	m_{10}	m_{11}

Hamming Codes: How They Work

- For m_6, 6 = 4 + 2, so we add to p_4 and p_2

	+1									
+1	+1		+1							
?	?	1	?	0	1	1	?	1	0	0
P_1	p_2	m_3	p_4	m_5	m_6	m_7	p_8	m_9	m_{10}	m_{11}

Hamming Codes: How They Work

- For m_7, $7 = 4 + 2 + 1$, so we add to p_4, p_2 and p_1

	+1									
+1	+1		+1							
+1	+1		+1							
?	?	1	?	0	1	1	?	1	0	0
P_1	p_2	m_3	p_4	m_5	m_6	m_7	p_8	m_9	m_{10}	m_{11}

Hamming Codes: How They Work

- What would we do for m_9?
  - $9 = 8 + 1$, so we add to p_8 and p_1

+1	+1									
+1	+1		+1							
+1	+1		+1				+1			
?	?	1	?	0	1	1	?	1	0	0
P_1	p_2	m_3	p_4	m_5	m_6	m_7	p_8	m_9	m_{10}	m_{11}

Hamming Codes: How They Work

- What would we do for m_{10}?
  - 10 = 8 + 2, so we add to p_8 and p_2
  - But m_{10} = 0, so we don't add anything

+1	+1									
+1	+1		+1							
+1	+1		+1				+1			
?	?	1	?	0	1	1	?	1	0	0
P_1	p_2	m_3	p_4	m_5	m_6	m_7	p_8	m_9	m_{10}	m_{11}

Hamming Codes: How They Work

- What would we do for m_{11}?
  - $11 = 8 + 2 + 1$, so we add to p_8, p_2 and p_1
  - But $m_{11} = 0$, so we don't add anything

+1	+1									
+1	+1		+1							
+1	+1		+1				+1			
?	?	1	?	0	1	1	?	1	0	0
p_1	p_2	m_3	p_4	m_5	m_6	m_7	p_8	m_9	m_{10}	m_{11}

Hamming Codes: How They Work

- Now we add up all of the parity bits
  - What would each one be? (Even parity)

+1	+1									
+1	+1		+1							
+1	+1		+1				+1			
?	?	1	?	0	1	1	?	1	0	0
P_1	p_2	m_3	p_4	m_5	m_6	m_7	p_8	m_9	m_{10}	m_{11}

Hamming Codes: How They Work

- Now we add up all of the parity bits
  - What would each one be? (Even parity)

+1	+1									
+1	+1		+1							
+1	+1		+1				+1			
1	?	1	?	0	1	1	?	1	0	0
P_1	p_2	m_3	p_4	m_5	m_6	m_7	p_8	m_9	m_{10}	m_{11}

Hamming Codes: How They Work

- Now we add up all of the parity bits
  - What would each one be? (Even parity)

+1	+1									
+1	+1		+1							
+1	+1		+1				+1			
1	1	1	?	0	1	1	?	1	0	0
P_1	p_2	m_3	p_4	m_5	m_6	m_7	p_8	m_9	m_{10}	m_{11}

Hamming Codes: How They Work

- Now we add up all of the parity bits
  - What would each one be? (Even parity)

+1	+1									
+1	+1		+1							
+1	+1		+1				+1			
1	1	1	0	0	1	1	?	1	0	0
P_1	p_2	m_3	p_4	m_5	m_6	m_7	p_8	m_9	m_{10}	m_{11}

Hamming Codes: How They Work

- Now we add up all of the parity bits
  - What would each one be? (Even parity)

+1	+1									
+1	+1		+1							
+1	+1		+1				+1			
1	1	1	0	0	1	1	1	1	0	0
P_1	p_2	m_3	p_4	m_5	m_6	m_7	p_8	m_9	m_{10}	m_{11}

Hamming Codes

- So we get 11100111100 as our bit string to send
- The receiver can recalculate the parity bits and make sure they match

Error Syndromes

The sender sent

11100111100

but what if we received

11000111100 ?

Can we correct this?

Error Syndromes

Recalculate parity bits and you get the numbers in blue:

0	0		0				1			
1	1	0	0	0	1	1	1	1	0	0
P_1	p_2	m_3	p_4	m_5	m_6	m_7	p_8	m_9	m_{10}	m_{11}

(We know there's an error because we didn't get 1101)

Error Syndromes

Add calculated parity bits to parity bits in received data:

=1	=1		=0				=0			
0	0		0				1			
1	1	0	0	0	1	1	1	1	0	0
p_1	p_2	m_3	p_4	m_5	m_6	m_7	p_8	m_9	m_{10}	m_{11}

Then reverse the sum and it will tell you the bit in error:

0011 -> third bit is wrong!

Now, you try! Decode this ASCII message (0b1000001 = 65 = ‘A’)

1011 0101 000

1111 0011 001

1100 0001 111

There may be bit errors! Assume each line encodes one byte of message data. (I.e., pad with a leading zero.)

Answers

1011 0101 000		1011 0100 000
1111 0011 001	Correct errors →	1111 0011 001
1100 0001 111		1110 0001 111

Extract message bits ↓

1010 000
1001 001
1000 111

← Convert to ASCII

PIG