Basics of Error Correcting Codes

Drawing from the book

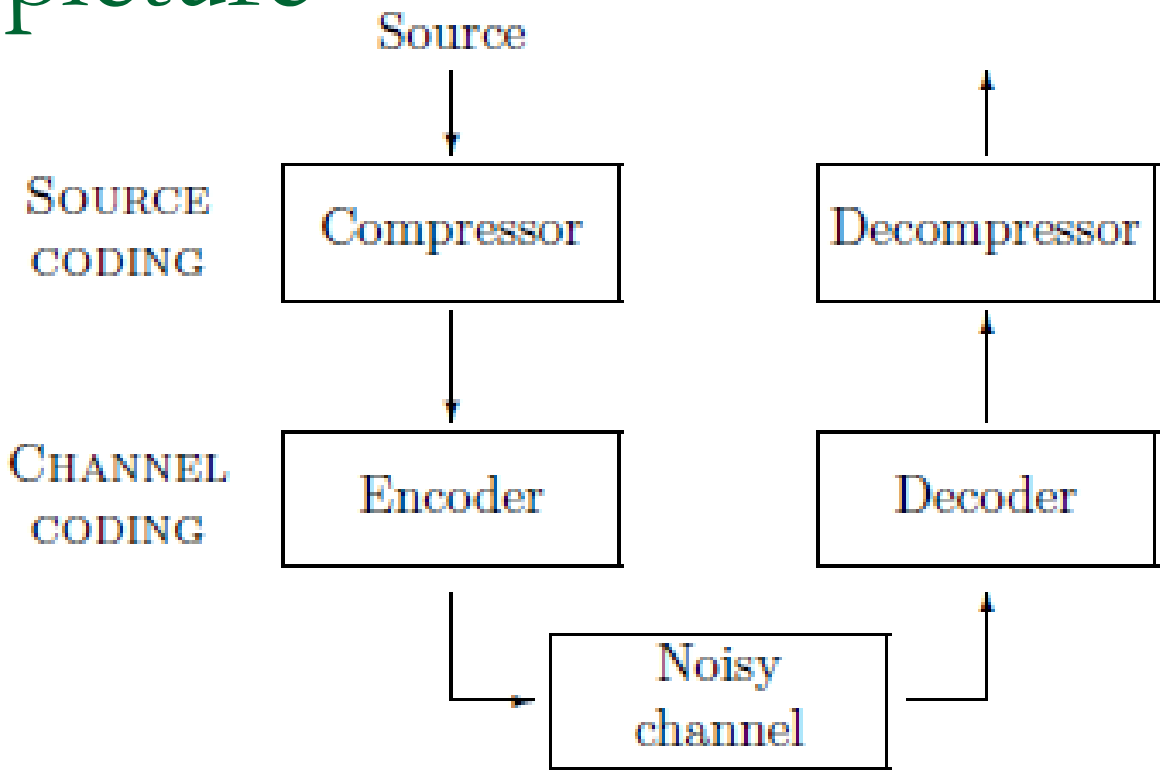
Information Theory, Inference, and Learning Algorithms
David MacKay
© Cambridge Univ. Press 2003

Downloadable or purchasable:
http://www.inference.phy.cam.ac.uk/mackay/itila/book.html

Channel coding aka Forward Error Correction

- "My communication system is working, but I am getting a lot of errors…what can I do?"
- CRC is an error DETECTING code…it spots errors with high probability, but doesn't tell you how to fix them
- Error CORRECTING codes can actually allow you to repair the errors…if there aren't too many

The big picture



- Channel coding is adding redundancy to improve reliability, at a cost in rate
  - Error correction
- Source coding is removal of redundancy from information bits to improve rate
  - Compression
- This lecture is only about channel coding

David MacKay
Information Theory, Inference, and Learning Algorithms
© Cambridge Univ. Press 2003

How do error correcting codes work?

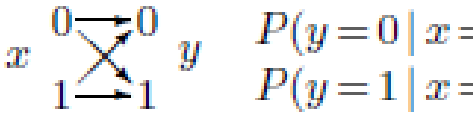
- Basic idea: add redundancy (extra bits) to make communication more robust
  - Or, put another way, don't allow all bit patterns, just a subset…if you receive an invalid bit sequence, correct to the closest valid bit sequence
- The extra bits (or disallowed bit patterns) reduce the net communication rate:
  - If "information bits" are denoted i and "error correction bits" denoted ec, then the new rate, with error correction is i/(i+ec)
  - The original rate, with no error correction (ec=0) is 1.0

Noisy communication channels

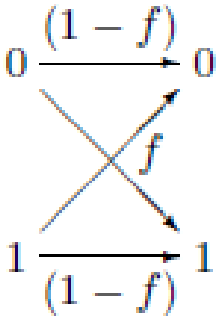
- EF modem → airgap → EF modem
- modem → phone line → modem
- wifi card → radio waves → wifi card
- Galileo probe → radio waves → Earth
- Parent cell → daughter cell 1
  → daughter cell 2
- RAM → disk drive → RAM
- RAM → flash memory → RAM
- printer → QR code → phone camera

A model for the noise in the channel

- Binary Symmetric Channel (BSC) with f=0.1
  - f: probability of bit ***f***lip



$$P(y=0\,|\,x=0) = 1-f; \quad P(y=0\,|\,x=1) = f;$$
$$P(y=1\,|\,x=0) = f; \quad P(y=1\,|\,x=1) = 1-f.$$



Other important channels: ***Erasure Channel*** (models packet loss in wired or wireless networks)

Example 1: Repetition code, "R3"

Received codeword	Decoded as
000	0 (no errors)
001	0
010	0
100	0
111	1 (no errors)
110	1
101	1
011	1

- Each 1 information bit gets encoded to 3 transmitted bits, so the rate of this code is 1/3
- If you think of the first bit as the message, and bits 2 and 3 as the error correction bits, then the rate also turns out to be 1/(1+2) = 1/3
- This code can correct 1 bit flip, or 2 bit erasures (erasures not shown)

Problems with R3

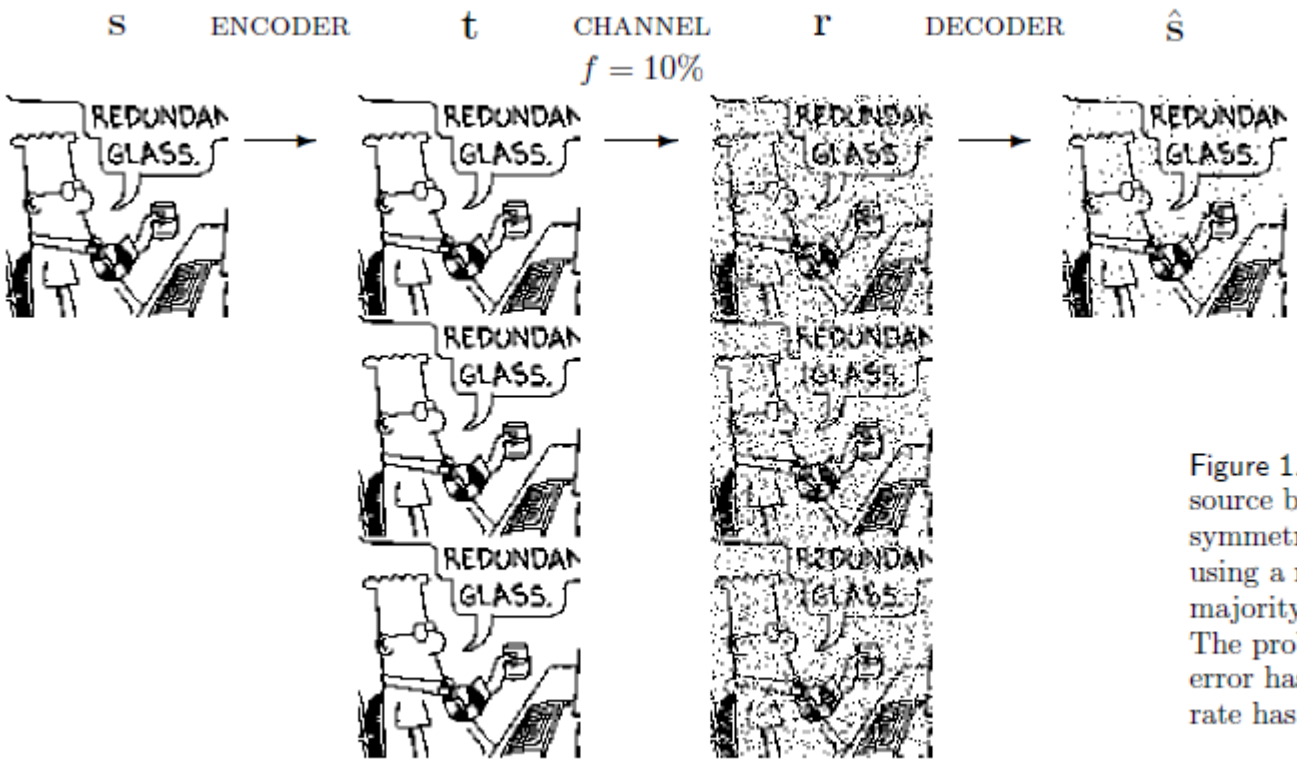


Figure 1.11. Transmitting 10 000 source bits over a binary symmetric channel with $f = 10\%$ using a repetition code and the majority vote decoding algorithm. The probability of decoded bit error has fallen to about 3%; the rate has fallen to 1/3.

Noise set to flip 10% of the bits
Rate is only 1/3
Still 3% errors remaining after error correction…Crummy!

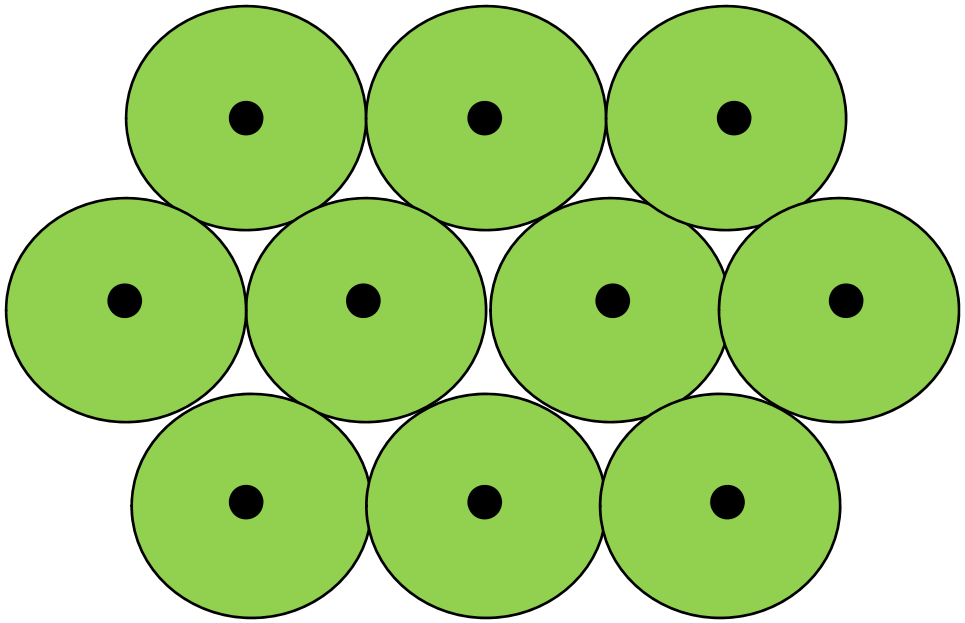
David MacKay
Information Theory, Inference, and Learning Algorithms
© Cambridge Univ. Press 2003

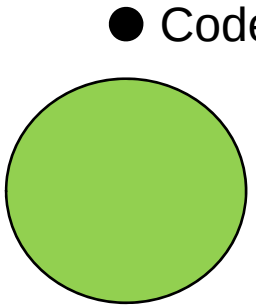
Example 2: Random code

Original message	Codewords transmitted
000	10100110
001	11010001
010	01101011
011	00011101
100	01101000
101	11001010
110	10111010
111	00010111

Each block of 3 info bits mapped to a random 8 bit vector…rate 3/8 code. Could pick any rate, since we just pick the length of the random code words. Note that we are encoding blocks of bits (length 3) jointly
Problems with this scheme:
(1) the need to distribute and store a large codebook
(2) decoding requires comparing received bit vectors to entire codebook

A visualization of ECCs





Codewords

Volume in which noise can (obviously) be tolerated

An error correcting code selects a subset of the space to use as valid messages (codewords). Since the number of valid messages is smaller than the total number of possible messages, we have given up some communication rate in exchange for robustness. The size of each ball above gives approximately the amount of redundancy. The larger the ball (the more redundancy), the smaller the number of valid messages

The name of the game

- In ECCs is to find mathematical schemes that allow time- and space-efficient encoding and decoding, while providing high communication rates and low bit error rates, despite the presence of noise

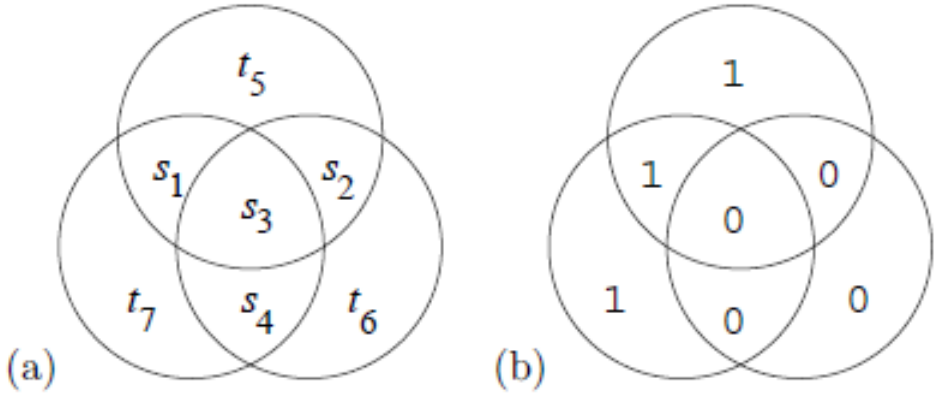
Types of ECC

- Algebraic
  - Hamming Codes
  - Reed-Solomon [CD, DVD, hard disk drives, QR codes]
  - BCH
- Sparse graph codes
  - Turbo [CDMA2000 1x]
  - Repeat accumulate
  - LDPC (Low Density Parity Check)
    - [WiMax, 802.11n, 10GBase 10 802.3an]
  - Fountain / Tornado / LT / Raptor (for erasure)
    - [3GPP mobile cellular broadcast, DVB-H for IP multicast]

Other ECC terminology

- Block vs. convolutional
- Linear
- Systematic / non-Systematic
  - Systematic means original information bits are transmitted unmodified.
    - Repetition code is systematic
    - Random code is not (though you could make a systematic version of a random code…append random check bits that don't depend on the data…would not be as good as parity bits that do depend on the data)

Example 3: (7,4) Hamming Code (Encoding)



b--example:
1000→1000101

Rate 4/7 code

Don't encode 1 bit at a time, as in the repetition code
Encode blocks of 4 source bits to blocks of 7 transmitted

$s_1s_2s_3s_4 \rightarrow t_1t_2t_3t_4t_5t_6t_7$

Where t_1 - t_4 are chosen s.t.

$s_1s_2s_3s_4 \rightarrow s_1s_2s_3s_4\, t_5t_6t_7$

Set parity check bits $t_5 – t_7$ using

$t_5=s_1+s_2+s_3 \mod 2 \rightarrow 1+0+0 = 1$
$t_6=s_2+s_3+s_4 \mod 2 \rightarrow 0+0+0 = 0$
$t_7=s_1+s_3+s_4 \mod 2 \rightarrow 1+0+0 = 1$

Parity check bits are a linear function information bits…a *linear code*

David MacKay
Information Theory, Inference, and Learning Algorithms
© Cambridge Univ. Press 2003

Example 3: (7,4) Hamming Code (Encoding)

The 16 codewords of the (7,4) Hamming code:

s	t	s	t	s	t	s	t
0000	0000000	0100	0100110	1000	1000101	1100	1100011
0001	0001011	0101	0101101	1001	1001110	1101	1101000
0010	0010111	0110	0110001	1010	1010010	1110	1110100
0011	0011100	0111	0111010	1011	1011001	1111	1111111

Any pair of codewords differs in at least 3 bits!

David MacKay
Information Theory, Inference, and Learning Algorithms

Example 3: (7,4) Hamming Code (Encoding)

Since it is a linear code, we can write the encoding operation as a matrix multiply (using mod 2 arithmetic):

t=$\mathbf{G}^T\mathbf{s}$ where

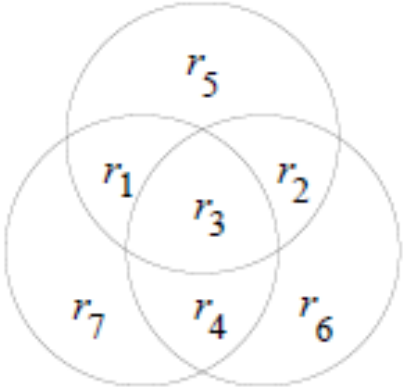
$$\mathbf{G}^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix},$$

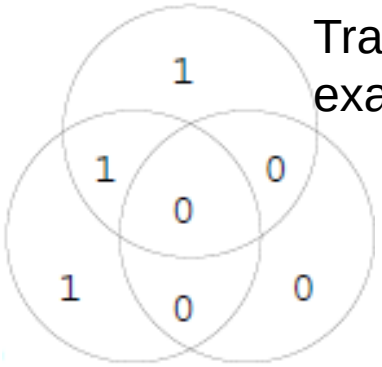
G is called the Generator Matrix of the code.

David MacKay
Information Theory, Inference, and Learning Algorithms

Example 3: (7,4) Hamming Code (Decoding)

If received vector r = t+n (transmitted plus noise), then write r in circles:

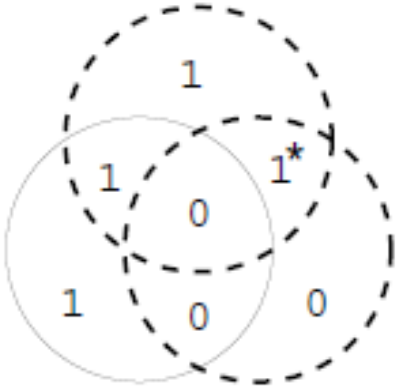


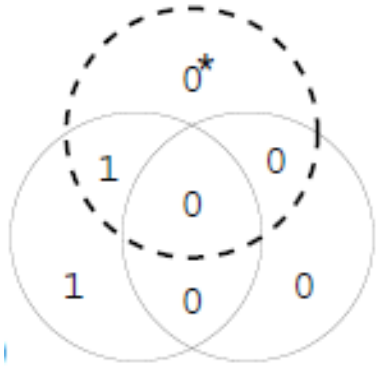


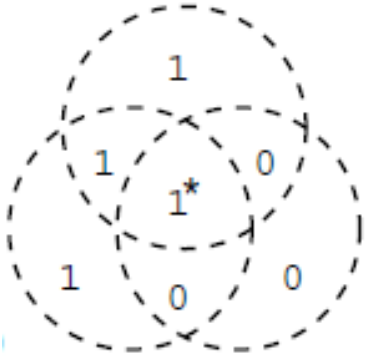
Transmitted example

Compute parity for each circle (dash➔violated parity check)
Pattern of parity checks is called the "syndrome"
Error bit is the unique one inside all the dashed circles

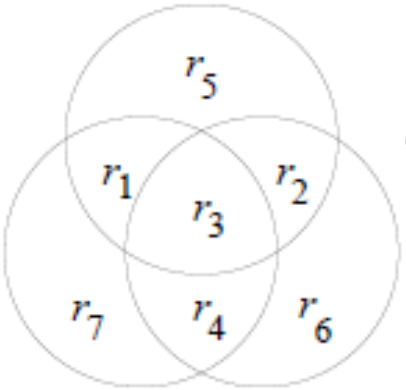
Dashed line➔parity check violated
* ➔bit flipped







Example 3: (7,4) Hamming Code (Decoding)



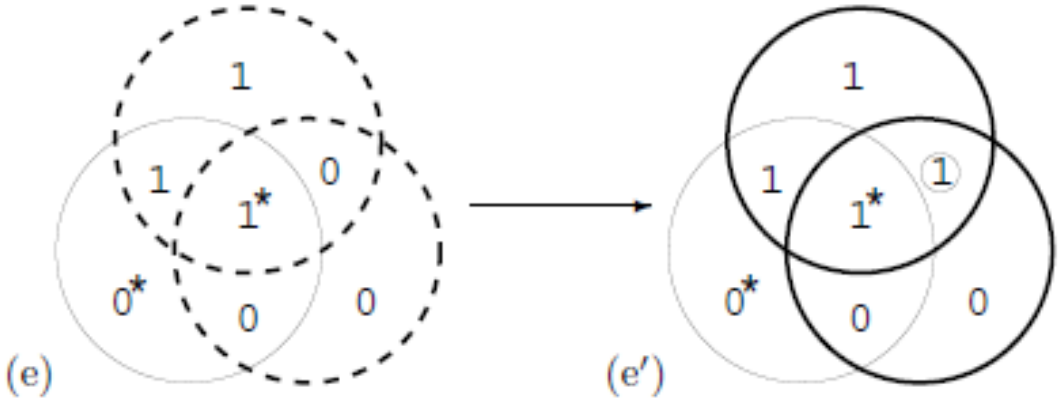
Each of the 3 circles is either dotted (syndrome=1) or solid (syndrome = 0)

➔ 2^3=8 possibilities

Syndrome z	000	001	010	011	100	101	110	111
Unflip this bit	*none*	r_7	r_6	r_4	r_5	r_1	r_2	r_3

David MacKay
Information Theory, Inference, and Learning Algorithms
© Cambridge Univ. Press 2003

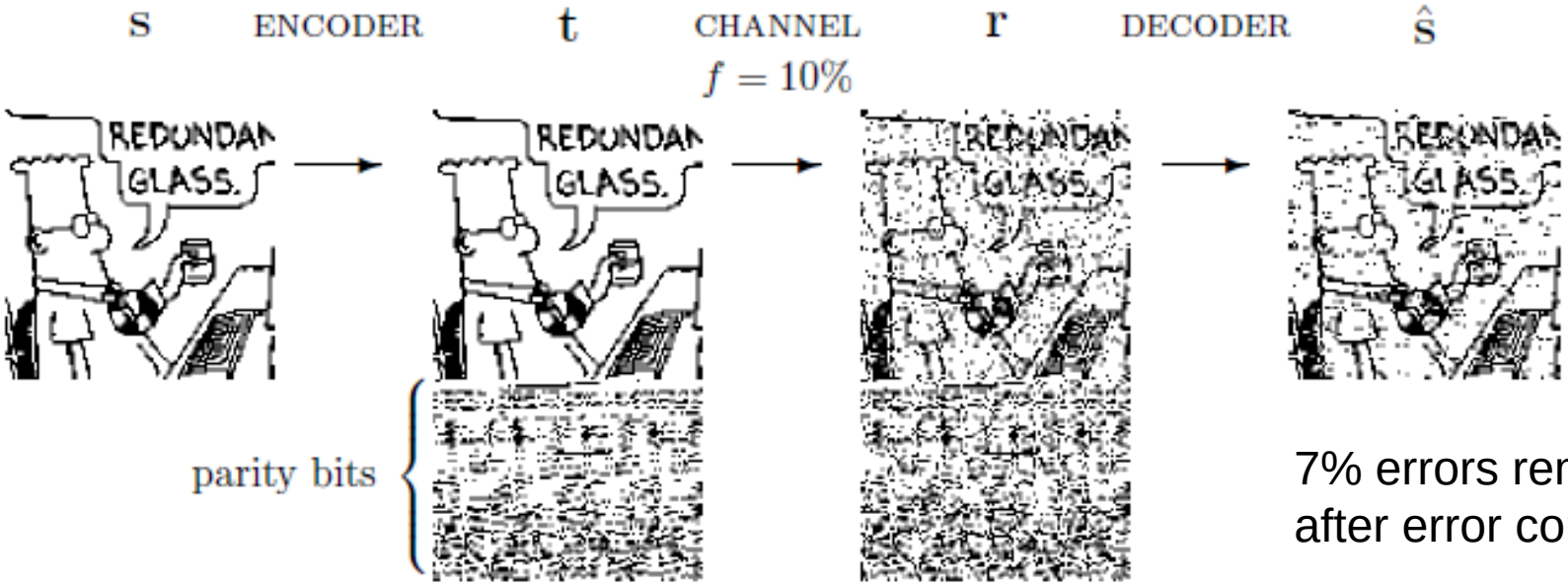
What happens if there are 2 errors?



*s denote actual errors
Circled value is incorrectly inferred single-bit error
Optimal decoder actually adds another error in this case…so we started with 2 errors and end with 3

David MacKay
Information Theory, Inference, and Learning Algorithms
© Cambridge Univ. Press 2003

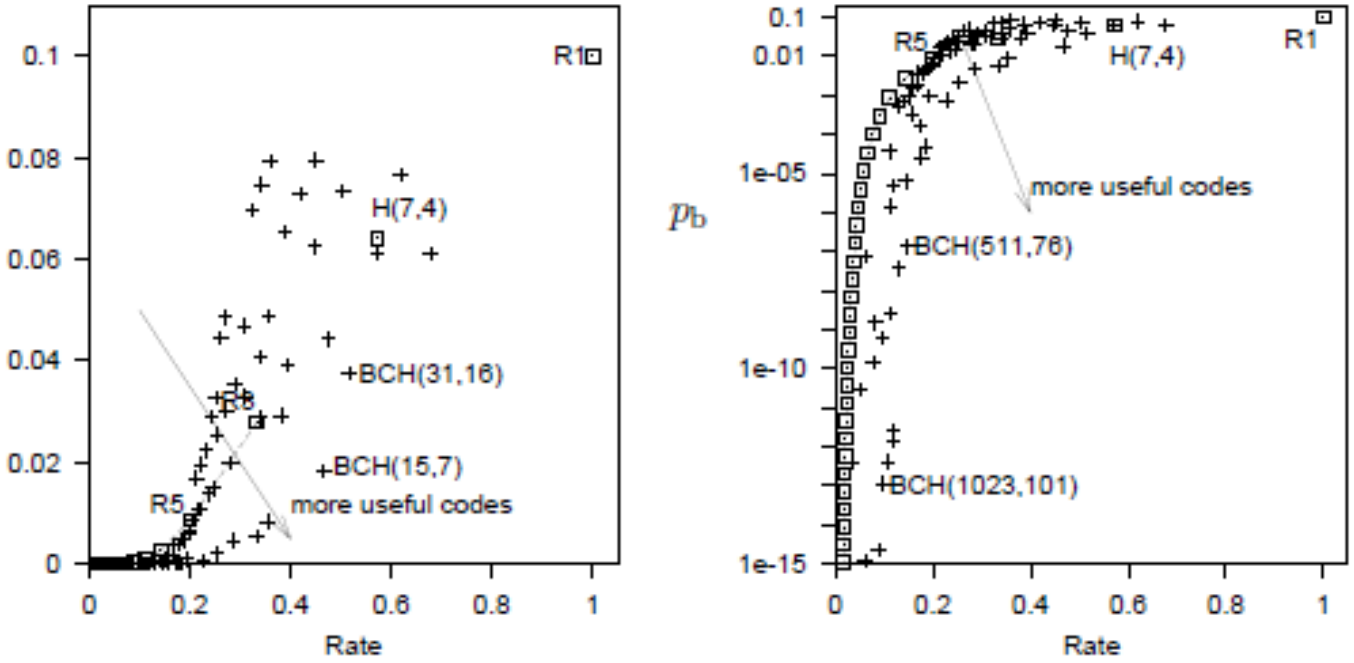
Larger (7,4) Hamming example



7% errors remain
after error correction

David MacKay
Information Theory, Inference, and Learning Algorithms
© Cambridge Univ. Press 2003

Comparing codes



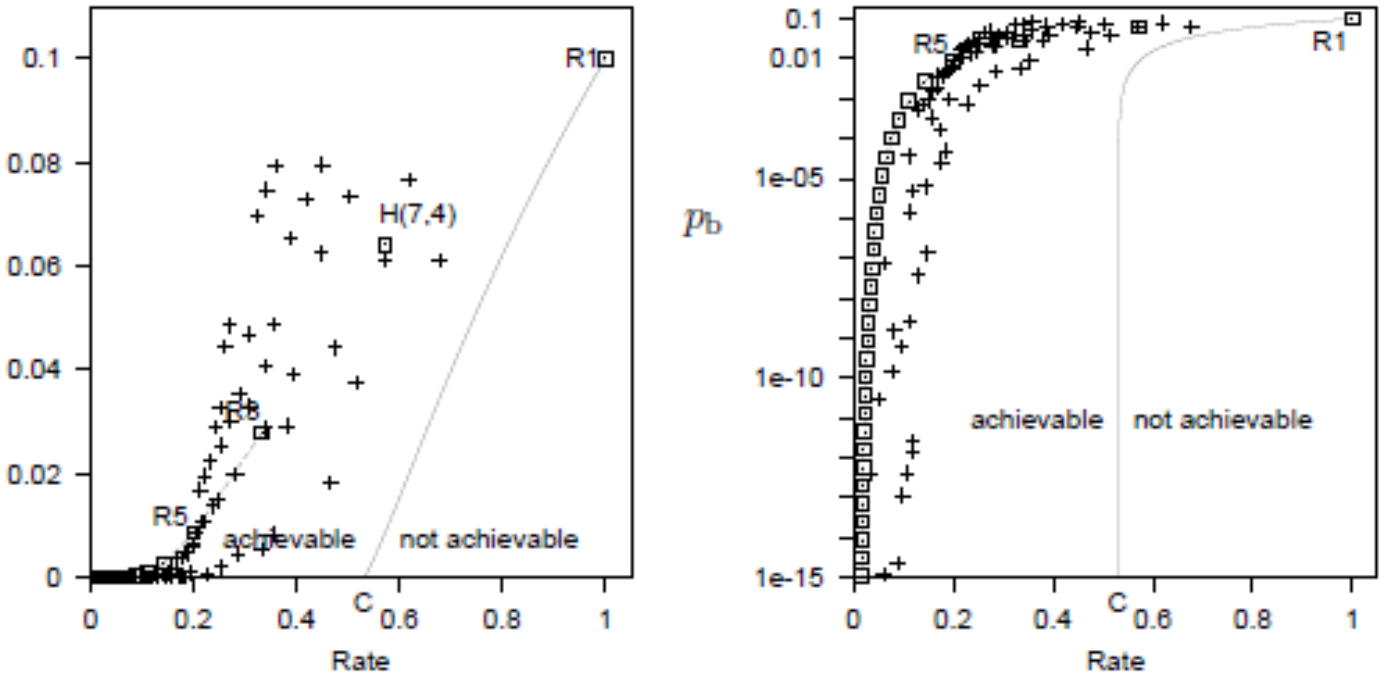
Binary symmetric channel with $f = 0.1$
Error probability p_b vs communication rate R for repetition codes, (7,4) Hamming code, BCH codes up to length 1023

David MacKay
Information Theory, Inference, and Learning Algorithms
© Cambridge Univ. Press 2003

What is the best a code can do?

- How much noise can be tolerated?
- What SNR do we need to communicate reliably?
- At what rate can we communicate with a channel with a given SNR?
  - What error rate should we expect?

What is the best a code can do?



- Binary symmetric channel with f =0.1

$$R = C / (1 - H_2(p_b))$$

where $H_2 = -p_b \log_2 p_b - (1-p_b)\log_2(1-p_b)$

David MacKay
Information Theory, Inference, and Learning Algorithms
© Cambridge Univ. Press 2003

End