

# Lecture 16a

error control coding

## Goals

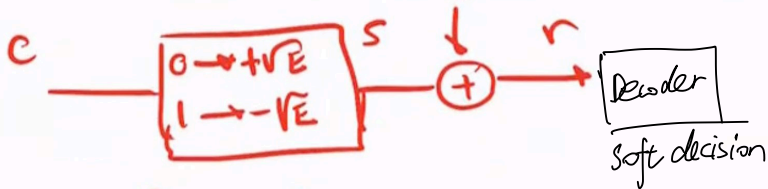
- Be able to encode using a linear block code
- Be able to decode a linear block code received over a binary symmetric channel or an additive white Gaussian channel

# Soft decisions vs Hard Decisions

Play 7 games:

Soft Decision ( Keep track of score  
Add total points to determine winner

Hard decision ( Just keep track of games won  
Team that won more games is winner



$$C = \{ \overset{c_0}{(000000)} \overset{c_1}{(111111)} \}$$

$$S_0 = (+VE, \dots, +VE)$$

$$S_1 = (-VE, \dots, -VE)$$

$$r = (100, 100, -1, -1, -1)$$

Opt receiver: find which signal is closest to the recd signal

$$C = \{ \overset{c_0}{(000000)} \overset{c_1}{(111111)} \}$$

$$\begin{aligned} s_0 &= (+\sqrt{E}, \dots, +\sqrt{E}) \\ s_1 &= (-\sqrt{E}, \dots, -\sqrt{E}) \end{aligned} \quad \text{) Equal energy}$$

$$r = (100, 100, -1, -1, -1)$$

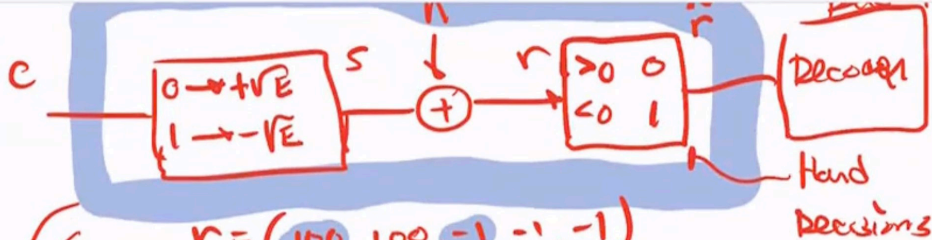
Opt receiver: find which signal is closest to the recd signal

Eqvv. find signal with largest correlation.

$$E=1 \quad (r, s_0) = 197 = (100+100+(-1)+(-1)+(-1))$$

$$(r, s_1) = -197$$

$\Rightarrow$  Receiver decides  $s_0$  ( $c_0$ )



$$r = (100, 100, -1, -1, -1)$$

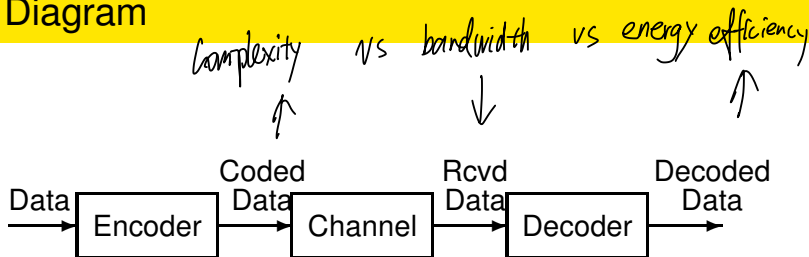
$$\hat{r} = (0, 0, 1, 1, 1) \quad \text{+}$$



Decode: find codeword closest to  $\hat{r}$

$\Rightarrow C_1 = (1, 1, 1, 1, 1)$  is output

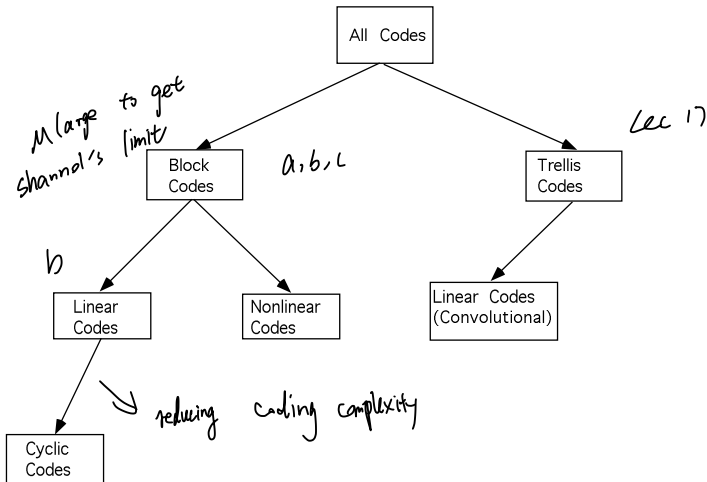
# Block Diagram



**Purpose:** To use large dimensional signal space to improve performance of a communication system. Usually this means trading increased complexity and increased bandwidth for significantly lower error probability at a given signal-to-noise ratio or equivalently lower transmitted energy for a given error probability.

# Taxonomy of Codes

*code is signal*



# Definitions

$$\begin{matrix} 1 & & M \\ \left[ \begin{matrix} v_1 \\ \vdots \\ v_n \end{matrix} \right] & \left[ \begin{matrix} v_1 \\ \vdots \\ v_n \end{matrix} \right] & \dots & \left[ \begin{matrix} v_1 \\ \vdots \\ v_n \end{matrix} \right] \end{matrix}$$

**Definition:** A block code  $C$  is a set of  $M$  vectors (of possible channel inputs) of length  $n$ .

**Definition:** The rate of a code is  $r = \frac{\log_2 M}{n}$  (measured in information bits/channel bit). *dimension*

**Definition:** The Hamming weight  $w_H(x)$  of a vector  $x$  is the number of nonzero components in  $x$ .

**Definition:** The minimum Hamming distance  $d_{H,\min}$  of a code is the smallest number of positions that any two (distinct) codewords differ by.

**Definition:** The minimum squared Euclidean distance  $d_{E,\min}^2$  of a code is the sum of the squares of the difference between code symbols from distinct codewords.

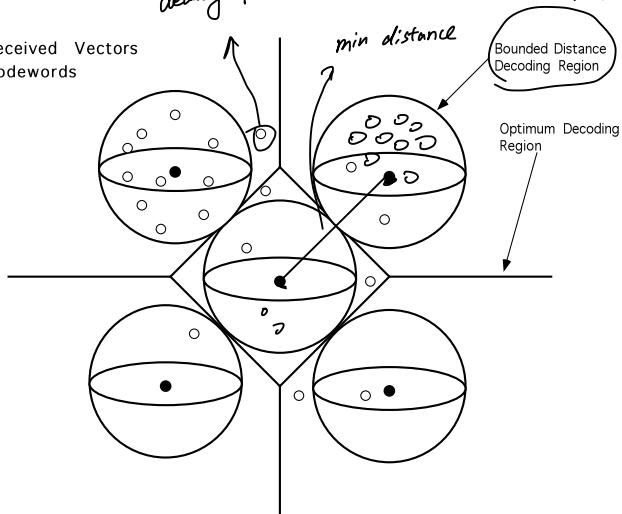


# Definitions

for bounded distance  
decoding failure

reduce 复杂度

- Received Vectors
- Codewords



# Representations

- There are usually two different representations of a code.
- In one representation the symbols are 0 and 1 while in the other representation the symbols are 1 and -1.
- The first representation is used when taking about adding two codewords (linearity) of a code.
- The second is used when talking about the Euclidean distance of a code. We use the mapping  $\tilde{b} = (-1)^b$  or

$$\begin{array}{rcl}
 b & \leftrightarrow & \tilde{b} \\
 \hline
 0 & \leftrightarrow & 1 \\
 1 & \leftrightarrow & -1
 \end{array}$$

# Examples

- Code 1 (Repetition Code).  $M = 2$ ,  $n = 5$ .
- Code 2:  $M = 4$ ,  $n = 4$ .
- Code 3: Cyclic Code,  $M = 4$ ,  $n = 9$ .
- Code 4: Hamming Code,  $M = 16$ ,  $n = 7$ .
- Code 5:  $M = 32$ ,  $n = 15$ .

# Code 1: Repetition Code, $M = 2$ , $n = 5$

$$\begin{aligned} C &= \{(0, 0, 0, 0, 0), (1, 1, 1, 1, 1)\}. \\ \hat{C} &= \{(1, 1, 1, 1, 1), (-1, -1, -1, -1, -1)\} \end{aligned}$$

This is called the repetition code. The rate is  $r = 1/n = 1/5$ . The minimum Hamming distance is  $d_{H,\min} = 5$ . The minimum squared Euclidean distance is

$$d_{E,\min}^2 = \sum_{k=1}^5 (1 - (-1))^2 = 5 \times 4 = 20.$$


## Code 2: $M = 4$ , $n = 4$

$$C = \{(0, 0, 0, 0), (0, 0, 1, 1), (1, 1, 0, 0), (1, 1, 1, 1)\}.$$

$$\hat{C} = \{(1, 1, 1, 1), (1, 1, -1, -1), (-1, -1, 1, 1), (-1, -1, -1, -1)\}$$

The rate of this code is  $1/2$ . The minimum Hamming distance of this code is 2. The minimum squared Euclidean distance is  $2 \times 4 = 8$ .

# Code 3: (Cyclic Code) $M = 4$ , $n = 9$

$$C = \{(0,0,0,0,0,0,0,0,0), (1,1,0,1,1,0,1,1,0) \\ (1,0,1,1,0,1,1,0,1), (0,1,1,0,1,1,0,1,1)\}$$


The rate of this code is  $2/9$ . The minimum Hamming distance of this code is 6. The minimum squared Euclidean distance is  $6 \times 4 = 24$ .

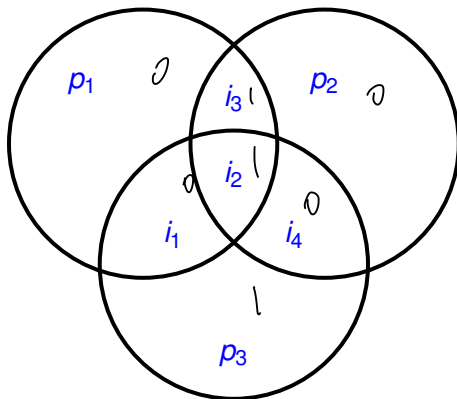
larger distance  $\Rightarrow$  lower rate

Code 4: (Hamming Code)  $M = 16$ ,  $n = 7$ 

?

$$C = \{(0, 0, 0, 0, 0, 0, 0), (0, 0, 0, 1, 0, 1, 1), \dots\}$$

even ones



$$C = (0110 \overset{p_1 p_2 p_3}{\underbrace{001}})$$

✓  
 $i_1, \dots, i_4$

# The Hamming Code

(from 0000000)

The codewords in this code are

7 code words distance 3  
 7 code words distance 4  
 1 code words dis 7

$d_{\min} = 3$

(0, 0, 0, 0, 0, 0, 0)

(1, 1, 1, 1, 1, 1, 1)

(1, 0, 0, 0, 1, 0, 1)

(0, 1, 0, 0, 1, 1, 1)

(1, 1, 0, 0, 0, 1, 0)

(1, 0, 1, 0, 0, 1, 1)

(0, 1, 1, 0, 0, 0, 1)

(1, 1, 0, 1, 0, 0, 1)

(1, 0, 1, 1, 0, 0, 0)

(1, 1, 1, 0, 1, 0, 0)

(0, 1, 0, 1, 1, 0, 0)

(0, 1, 1, 1, 0, 1, 0)

(0, 0, 1, 0, 1, 1, 0)

(0, 0, 1, 1, 1, 0, 1)

(0, 0, 0, 1, 0, 1, 1)

(1, 0, 0, 1, 1, 1, 0)

from this codeword

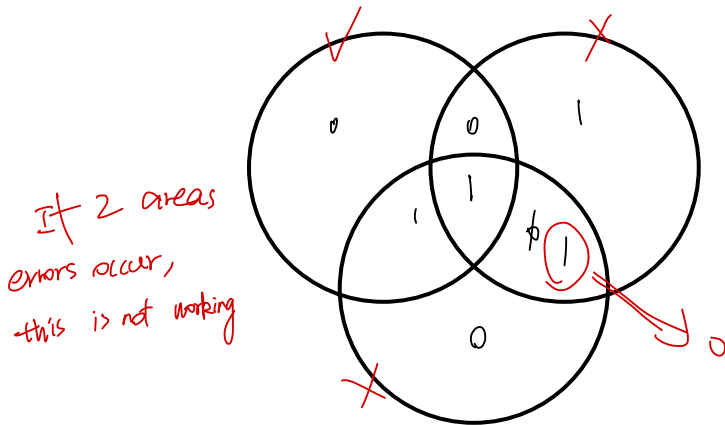
7 dist 3  
 7 dist 4  
 1 dist 7

)  $d=3$



# Code 4: (Hamming Code) $M = 16$ , $n = 7$

$$C = \{(0, 0, 0, 0, 0, 0, 0), (0, 0, 0, 1, 0, 1, 1), \dots\}$$



# Parity checks equations for Hamming code

$$+1 = -1 \pmod{2}$$

$$p_1 = i_1 + i_2 + i_3 \pmod{2}$$

$$p_2 = i_2 + i_3 + i_4 \pmod{2}$$

$$p_3 = i_1 + i_2 + i_4 \pmod{2}$$

$$i_1 + i_2 + i_3 + p_1 = 0 \pmod{2}$$

$$i_2 + i_3 + i_4 + p_2 = 0 \pmod{2}$$

$$i_1 + i_2 + i_4 + p_3 = 0 \pmod{2}$$

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

↗ code words

# Parity checks equations for Hamming code

Let  $\mathbf{c} = (i_1, i_2, i_3, i_4, p_1, p_2, p_3)$  and

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

Then we obtain the relation  $H\mathbf{c}^T = \mathbf{0}^T$ .

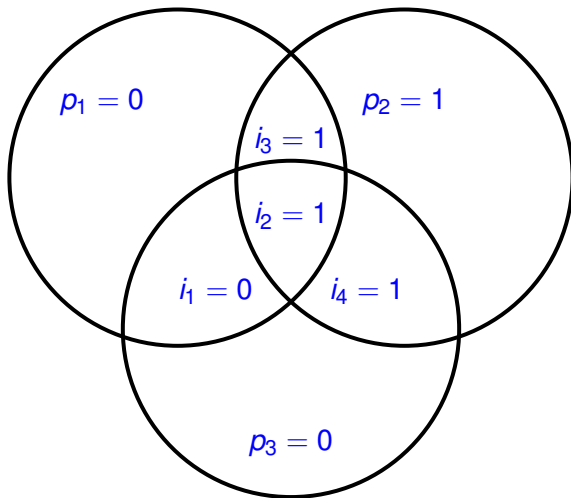
# Parity checks equations for Hamming code

The way the Hamming code works is as follows. There are four information bits from which three additional parity bits are determined (as described below). These 7 bits are transmitted over the channel. The channel (modulator, waveform channel, demodulator) may make errors.

## Encoding

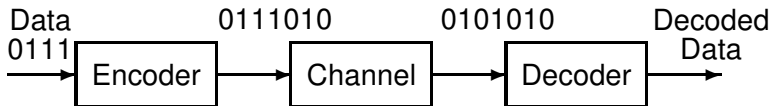
There are seven regions in the circles. There are four regions in two or more circles. In these four regions the information bits are written ( $i_1, i_2, i_3, i_4$ ). The three parity bits ( $p_1, p_2, p_3$ ) are determined by making sure that the number of ones in each circle is even. The seven bits are transmitted over the channel. For example, if  $i_1 = 0, i_2 = 1, i_3 = 1, i_4 = 1$  then the number of ones in the left circle (without including  $p_1$ ) is 2 which is even so  $p_1 = 0$ . Similarly  $p_2 = 1$  in order to make the number of ones in the right circle even. Finally  $p_3 = 0$ . So the codeword transmitted is (0111010).

# Code 4: (Hamming Code) $M = 16$ , $n = 7$ .



# Channel Effects

The channel may make an error in one of the bits.

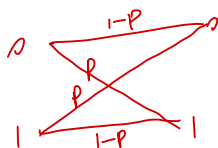


$$r = \frac{4}{7} \leftarrow \begin{array}{l} 4 \text{ bits} \\ 7 \text{ times} \end{array}$$

For example, the previous codeword could be received as (0101010).

$$d_{H \min} = 3$$

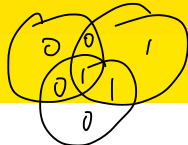
Trans



rece

Binary symmetric channel

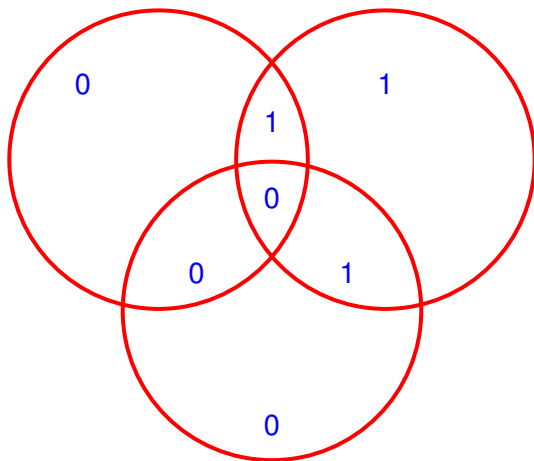
# Decoding



The decoder does not know which bit (if any) was changed by the channel. The decoder recomputes the parity and finds out which of the three circles has the correct parity. The error (if there is one or less) occurred in the circles where the parity did not check but not in the circles where the parity did check.

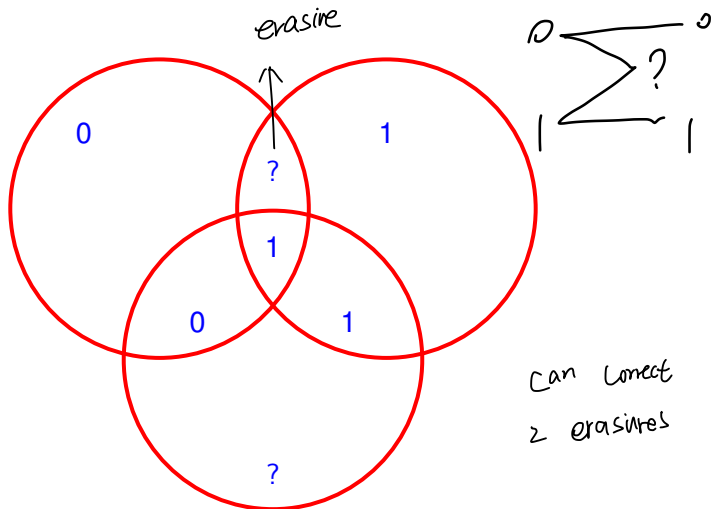
For the received vector being (0101010) we recompute the parity based on the information bits being (0101) and the parity bits being (010). In this case the parity of the left circle is odd. The parity of the right circle is odd and the parity of the bottom circle is even. Thus we conclude that a single error must have occurred in the two upper circles but not the lower circle and thus the bit representing  $i_3$  was in error. The decoder would put out the decoded information (0111). This decoder can correct any single error. It can also correct any pattern of two erasures.

## Code 4: Decoding the Hamming Code





# Code 4: Decoding the Hamming Code



## Code 5

$$r = \frac{\log_2 M}{N}$$

$M = 32$ ,  $n = 15$ ,  $d_{H,min} = 7$ ,  $r = \log_2(32)/15 = 1/3$

000000000000000	111111111111111
000111011001010	000100110101111
001110110010100	001001101011110
011101100101000	010011010111100
111011001010000	100110101111000
110110010100001	001101011110001
101100101000011	011010111100010
011001010000111	110101111000100
110010100001110	101011110001001
100101000011101	010111100010011
010100001110110	101111000100110
101000011101100	011110001001101
010000111011001	111100010011010
100001110110010	111000100110101
000011101100101	110001001101011
001010000111011	100010011010111

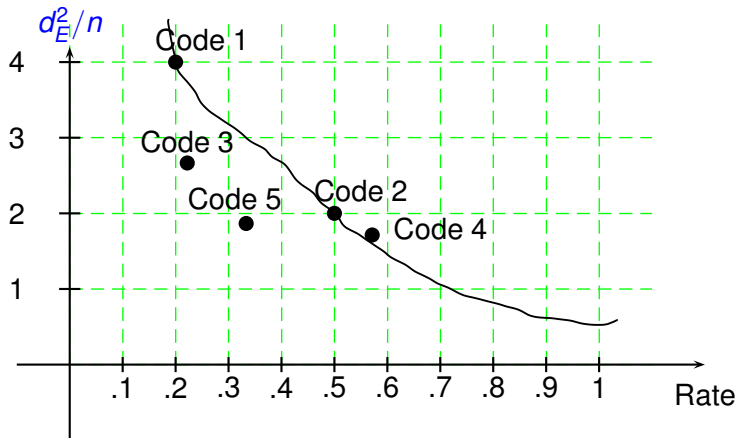
## Code Comparison

lower  $R \downarrow$ ,  $d_H \uparrow$

$$d_E^2 = 4 d_H$$

	Example	k	n	Rate	$d_E^2$	$d_H$	$d_{Hr}$
rep $\rightarrow$	1	1	5	1/5	20	5	1
	2	2	4	1/2	8	2	1
	3	2	9	2/9	24	6	1.33
	4	4	7	4/7	12	3	1.71
Hamming co $\rightarrow$	5	5	15	1/3	28	7	2.33

# Code Comparison



# Coding and Modulation

For binary coding and antipodal modulation (e.g. BPSK) the relation between Euclidean distance and Hamming distance is

$$\boxed{d_{E,\min}^2 = 4Ed_{H,\min}} \quad (\sqrt{E} - (-\sqrt{E}))^2 = 4E$$

where the signal values take values  $\{\pm\sqrt{E}\}$ .

The relation between energy per coded bit  $E$  and energy per information bit,  $E_b$ , is

energy per dimension  $\rightarrow$   $E = E_b \log_2(M)/n = E_b r.$

Thus the end result is that the Euclidean distance is

$$d_{E,\min}^2 = 4E_b r d_{H,\min}.$$

want this product large

# Coding and Modulation

The goal in designing a code is to have as many codewords as possible (large  $M$ ) but also as far apart as possible. There is a tradeoff between the code rate and the Euclidean distance of a code. The larger the code rate the smaller the Euclidean distance and the smaller the code rate the larger the Euclidean distance of a code.

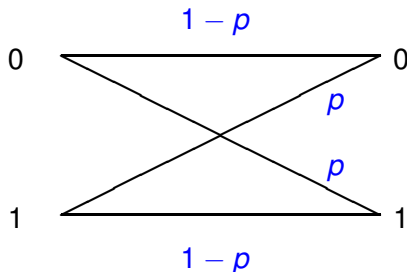
# Coding and Modulation

The encoder maps information symbols into codewords. For Code 2 the encoder has the following mapping.

$$\begin{aligned}00 &\rightarrow (1, 1, 1, 1) \\01 &\rightarrow (1, 1, -1, -1) \\10 &\rightarrow (-1, -1, 1, 1) \\11 &\rightarrow (-1, -1, -1, -1)\end{aligned}$$

The decoder must map the received symbols from the channel (which may be distorted) back to information symbols. It should do this to minimize the probability of making an error.

# Example 1: Code 1, Binary Symmetric Channel



*majority vote*  
Can correct 2 errors

Data	Coded Data	Rcvd Data	Decoded Data
0	00000	01010	0
1	11111	11100	1

(Codewords of length  $n$ ).

Decoding Rule: Choose codeword “closest” to demodulated data.

Conclusion: Decoder can correct up to  $\lfloor (n-1)/2 \rfloor = 2$  errors with very

low complexity.



# Error Correcting Capability

## Proposition:

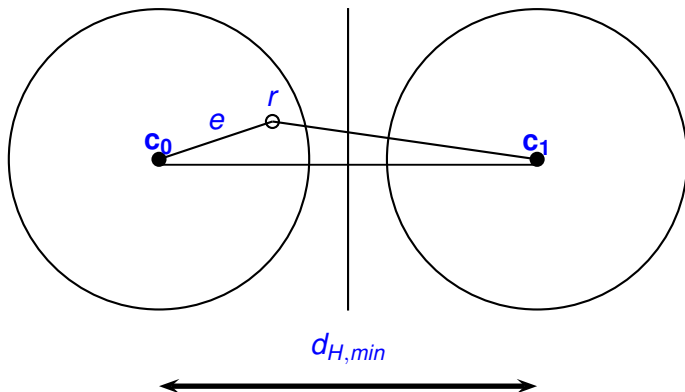
A block code (when used on a binary symmetric channel) can correct  $e$  errors provided  $2e + 1 \leq d_{H,min}$ .

**Proof:** The decoder chooses the codeword that is closest in Hamming distance to the received vector. If  $e$  errors occurred then the received vector is distance  $e$  from the transmitted codeword. If the errors are in the right positions then the received vector could be distance  $d_{H,min} - e$  from a codeword at the minimum distance. So correct decoding will occur if

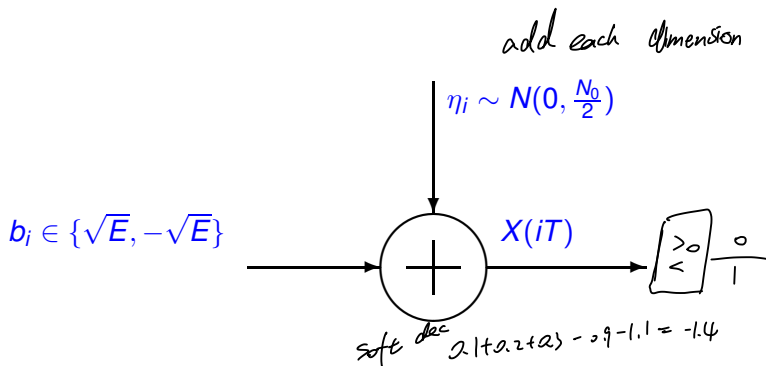
$$\begin{aligned} e &< d_{H,min} - e \\ 2e &< d_{H,min} \\ 2e + 1 &\leq d_{H,min} \end{aligned}$$

# Error Correcting Capability

$$e < \frac{d_{H,min}}{2}$$



# Example 2: Code 1 (Repetition Code) and an AWGN channel

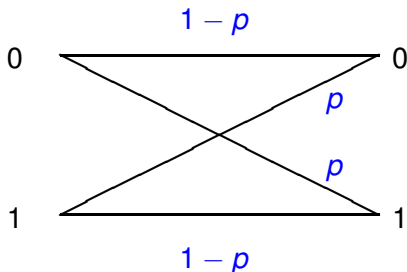


The decoder finds the transmitted codeword by finding the coded data that is closest in Euclidean distance to the received data.

Data	Coded Data	Rcvd Data	Decoded D
0	$(+\sqrt{E}, +\sqrt{E}, +\sqrt{E}, +\sqrt{E}, +\sqrt{E})$	1.2 0.5 0.8 -0.1 0.9	0
1	$(-\sqrt{E}, -\sqrt{E}, -\sqrt{E}, -\sqrt{E}, -\sqrt{E})$	0.1 0.2 0.3 -0.9 -1.1	1

# Example 3: Code 2 Binary Symmetric Channel

*can detect error*



Data	Coded Data	Rcvd Data	Decoded Data
00	0000	0000	00
01	0011	0111	11

In the second case the received vector is equally close to two different codewords and chooses the wrong codeword.

# Example 4. Code 2 and an AWGN Channel

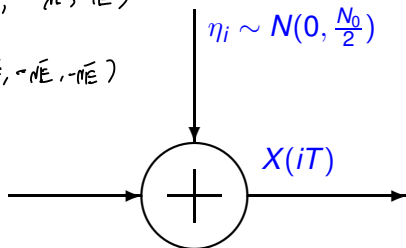
$$0000 \rightarrow (+\sqrt{E}, \dots, +\sqrt{E})$$

$$0011 \rightarrow (+\sqrt{E}, +\sqrt{E}, -\sqrt{E}, -\sqrt{E})$$

$$\vdots$$

$$1111 \rightarrow (-\sqrt{E}, -\sqrt{E}, -\sqrt{E}, -\sqrt{E})$$

$$b_i \in \{+\sqrt{E}, -\sqrt{E}\}$$



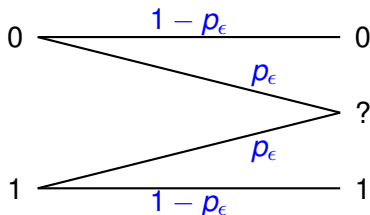
The decoder finds the transmitted codeword by finding the coded data that is closest in Euclidean distance to the received data.

Data	Coded Data	Rcvd Data	Decoded
00	$(+\sqrt{E}, +\sqrt{E}, +\sqrt{E}, +\sqrt{E})$	1.2 0.5 -0.1 0.9	00
01	$(+\sqrt{E}, +\sqrt{E}, -\sqrt{E}, -\sqrt{E})$	0.1 0.2 -0.9 -1.1	01

# Example 5. Code 1 (Repetition Code) and Binary Erasure Channel

$$m = 32$$

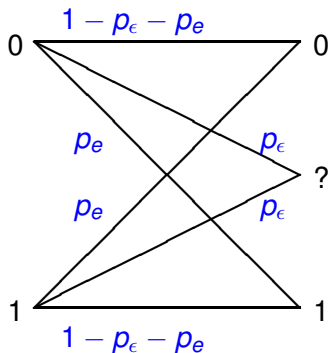
$$n = 15$$



Data	Coded Data	Rcvd Data	Decoded Data
0	00000	0?0?0	0
1	11111	110??	Not possible

The codewords are of length  $n$ . The decoder can correct up to  $n - 1$  erasures.

# Example 6. Code 1 (Repetition Code) and Binary Errors and Erasures Channel



## Example 6. Code 1 (Repetition Code) and Binary Errors and Erasures Channel

Data	Coded Data	Rcvd Data	Decoded Data
0	00000	0?0?1	0
1	11111	110??	1

Codewords of length  $L$ . Decoder ignores erased position and chooses closest “decimated” codeword. Decoder can correct  $e$  errors and  $\tau$  erasures if

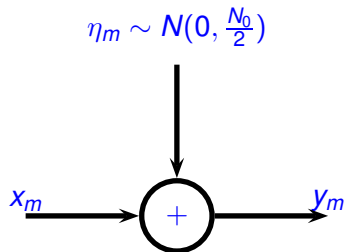
$$2e + \tau \leq n - 1.$$

For other codes the decoder can correct  $e$  errors and  $\tau$  erasures provided  $2e + \tau \leq d_{min} - 1$  where  $d_{min}$  is the minimum Hamming distance between codewords.

Thus a code can correct (fill) twice as many erasures as it can correct errors. With an erasure the location of the disturbance is known, whereas with an error the decoder does not know where it occurred.

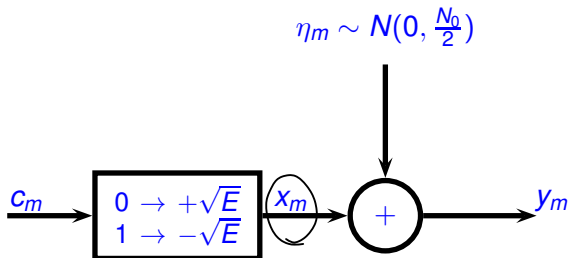


# Additive white Gaussian noise (AWGN) channel



- Since many codes used in practice are binary codes with code symbols being just 0 or 1, we need a way to map the code symbols (coded bits) into real numbers for transmission over the AWGN channel.
- For this purpose a mapping is added that maps coded bits  $c_m$  with alphabet  $\{0, 1\}$  to  $+\sqrt{E}$  and  $-\sqrt{E}$  as shown below
- This channel has binary input and real numbers as the output. A decoder that can process real numbers is sometimes called a “soft decision decoder.”

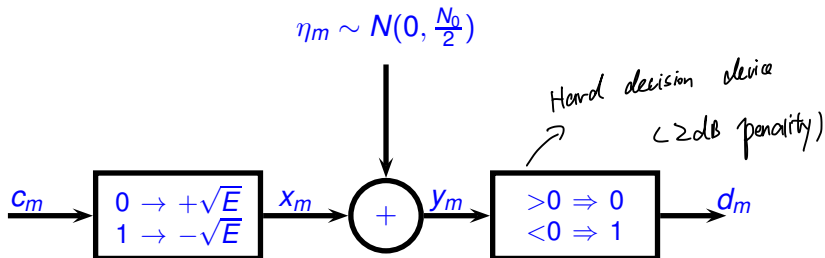
# Binary input additive white Gaussian noise (BI-AWGN) channel



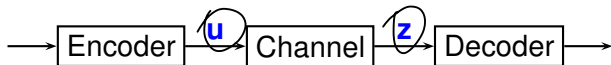
- Some codes only have practical decoding algorithms when using bits at the input of the decoder (channel output).
- As such, for these codes it is necessary to make a hard decision about each coded bit and then the decoder is said to be a hard decision decoder.
- This is shown below. In this case the channel is the same as a binary symmetric channel with crossover probability  $p = Q(\sqrt{2E/N_0})$ .

# Binary input, binary output additive white Gaussian noise (BIBO-AWGN) channel

$$p = Q\left(\sqrt{\frac{2E}{N_0}}\right)$$



# Minimum Error Probability and Maximum Likelihood Decoding



Given a channel transition rule  $p(\mathbf{z}|\mathbf{u})$  between the input  $\mathbf{u} = (u_0, u_2, \dots, u_{n-1})$  to a (discrete) channel and the output  $\mathbf{z} = (z_0, z_1, z_2, \dots, z_{n-1})$  and a set of possible transmitted (input) vectors (a code)  $\mathcal{C} = \{\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_{M-1}\}$  with  $\mathbf{c}_i = (c_{i,0}, c_{i,1}, \dots, c_{i,n-1})$  we would like to find the optimal rule for determining, based on observing  $\mathbf{z}$ , which of the  $M$  vectors from  $\mathcal{C}$  was transmitted. The goal is to minimize the overall probability of making an error in our decision.

# Minimum Error Probability and Maximum Likelihood Decoding

Let  $H_m$  denote the event that the input to the channel (the codeword selected) is  $\mathbf{c}_m$ . Let  $\pi_m$  be the probability of this event; that is  $\pi_m = P\{H_m\}$ . The decoder observes  $\mathbf{z}$  and makes a decision on which of the events  $H_m$ ,  $m = 0, 1, 2, \dots, M-1$  occurred. The probability of error is given by

$$\begin{aligned}
 P_e &= \sum_{m=0}^{M-1} \sum_{k=0, k \neq m}^{M-1} P\{\text{decoder decides } H_k | H_m\} P\{H_m\} \\
 &= \sum_{m=0}^{M-1} \left[ \sum_{k=0, k \neq m}^{M-1} P\{\text{decide } H_k | H_m\} \right] \pi_m \\
 &= \sum_{m=0}^{M-1} [1 - P\{\text{decide } H_m | H_m\}] \pi_m
 \end{aligned}$$

# Minimum Error Probability and Maximum Likelihood Decoding

Consider the quantity  $P\{\text{decide } H_m | H_m\}$ . Let  $R_m$  be the set of all (channel) output vectors for which the decoder decides event (hypothesis)  $H_m$  occurred. Event  $H_m$  occurring is equivalent to the event that  $\mathbf{c}_m$  was the input to the channel. Then

$$P\{\text{decide } H_m | H_m\} = \int_{R_m} p(\mathbf{z} | \mathbf{H}_m) d\mathbf{z}$$

where the above integral is interpreted in one of two ways. If the channel is a continuous output channel then  $p(\mathbf{z} | \mathbf{H}_m)$  is a conditional density function and the integral is a multidimensional integral over the region  $R_m$ . If the channel is a finite output channel then  $p(\mathbf{z} | \mathbf{H}_m)$  is a conditional probability mass function and the integral is really a sum over the region  $R_m$ . This will be made clear in the examples given below. Since  $p(\mathbf{z} | \mathbf{H}_m) = p(\mathbf{z} | \mathbf{c}_m)$  the above expression for probability of error becomes



# Minimum Error Probability and Maximum Likelihood Decoding

$$\begin{aligned}
 P_e &= \sum_{m=0}^{M-1} \pi_m - \sum_{m=0}^{M-1} \int_{R_m} p(\mathbf{z}|\mathbf{c}_m) \pi_m d\mathbf{z} \\
 &= 1 - \sum_{m=0}^{M-1} \int_{R_m} p(\mathbf{z}|\mathbf{c}_m) \pi_m d\mathbf{z}.
 \end{aligned}$$

To minimize the average error probability we would like to choose the regions  $R_m$  to maximize the second term. If for a given channel output,  $p(\mathbf{z}|\mathbf{c}_5)\pi_5$  is larger than  $p(\mathbf{z}|\mathbf{c}_k)\pi_k$  for  $k \neq 5$  then choosing  $\mathbf{z}$  to be in  $R_5$  would make the last term largest. Thus the decision rule that minimizes average error probability is

$$\mathbf{z} \in \mathbf{R}_m \text{ if } p(\mathbf{z}|\mathbf{c}_m)\pi_m = \max_{0 \leq k \leq M-1} p(\mathbf{z}|\mathbf{c}_k)\pi_k.$$

# Minimum Error Probability and Maximum Likelihood Decoding

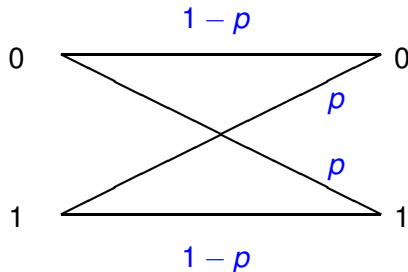
This is called the *maximum a posteriori probability* (MAP) decision rule. It is the rule that minimizes the average error probability. If all the prior probabilities are identical  $\pi_m = 1/M$  then the optimum decision rule then reduces to

$$\mathbf{z} \in \mathbf{R}_m \text{ if } p(\mathbf{z}|\mathbf{c}_m) = \max_{1 \leq k \leq M} p(\mathbf{z}|\mathbf{c}_k). \quad \text{MAP}$$

This is called the *maximum likelihood decoding* rule. It is useful if the prior probabilities are unknown to the system designer. In a digital communications context, if proper source coding has been done then the distribution on the input symbols should be uniform so maximum likelihood decoding optimum.

# Example 1

Consider a binary symmetric channel with crossover probability  $p < 1/2$ . The input and output alphabet is just  $\{0, 1\}$ .



$$p(1|0) = p$$

$$p(0|1) = p$$

$$p(0|0) = 1-p$$

$$p(1|1) = 1-p$$

$$p(00|01) = \frac{(1-p)p}{\downarrow}$$

$\downarrow$

independent

(also memory less channel) errors

$$\mathbf{u} = (u_0, u_1, \dots, u_{n-1})$$

$$\mathbf{z} = (z_0, z_1, \dots, z_{n-1})$$

# Example 1

The transition probability between an input vector and an output vector is

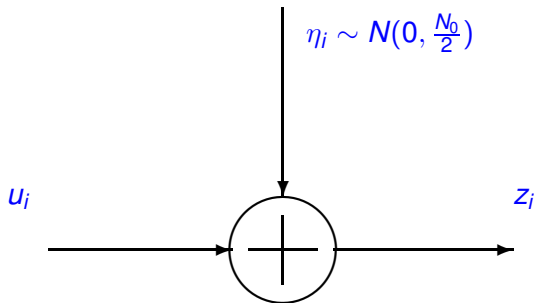
$$p(\mathbf{z}|\mathbf{u}) = p^{d_H(\mathbf{z},\mathbf{u})}(1-p)^{n-d_H(\mathbf{z},\mathbf{u})} = \left[\frac{p}{1-p}\right]^{d_H(\mathbf{z},\mathbf{u})}(1-p)^n$$

where  $d_H(\mathbf{z}, \mathbf{u})$  is the Hamming distance between the input vector  $\mathbf{u}$  and the output vector  $\mathbf{z}$ . If  $p < 1/2$  then  $\frac{p}{1-p} < 1$ . So maximizing  $p(\mathbf{z}|\mathbf{u})$  is equivalent to minimizing  $d_H(\mathbf{z}, \mathbf{u})$ . So for a binary symmetric channel the optimum decision rule (assuming equal a priori probabilities) is to choose  $H_m$  (or equivalently  $\mathbf{c}_m$ ) if  $d_H(\mathbf{z}, \mathbf{c}_m)$  is smallest. That is, choose the codeword closest in Hamming distance to the received vector (channel output).



## Example 2


Consider an additive white Gaussian noise channel.



## Example 2

The noise  $n_i$  is Gaussian, mean 0 and variance  $N_0/2$ . The possible inputs to the channel are a finite set of real numbers (e.g.  $u_i \in \{+\sqrt{E}, -\sqrt{E}\}$ ). The transition probability is

$$\begin{aligned} \underline{p(\mathbf{z}|\mathbf{u})} &= \prod_{i=0}^{n-1} \frac{1}{\sqrt{2\pi\sigma}} \exp\left\{-\frac{1}{2\sigma^2}(z_i - u_i)^2\right\} = \left[\frac{1}{\sqrt{2\pi\sigma}}\right]^n \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=0}^{n-1} (z_i - u_i)^2\right\} \\ &= \left[\frac{1}{\sqrt{2\pi\sigma}}\right]^n \exp\left\{-\frac{1}{2\sigma^2} d_E^2(\mathbf{z}, \mathbf{u})\right\}. \end{aligned}$$

$d_E^2(\mathbf{z}, \mathbf{u}) = \|\mathbf{z}\|^2 - 2(\mathbf{z}, \mathbf{u}) + \|\mathbf{u}\|^2$   


where  $d_E^2(\mathbf{z}, \mathbf{u}) = \sum_{i=0}^n (\mathbf{z}_i - \mathbf{u}_i)^2$  is the squared Euclidean distance between the channel input and output. Thus the optimal decoder finds  $\mathbf{u}$  to minimize  $d_E^2(\mathbf{z}, \mathbf{u})$ . If the transmitted signals or codewords have equal energy then finding the codeword with the smallest Euclidean distance is the same as finding the codeword with the largest correlation ( $\sum_{i=1}^N z_i u_i$ ).

# Geometry of Repetition Codes

$$n=5$$

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} = 16$$

$$1 + 5 + 10 = 16$$

- Code 1 (the repetition code) has 2 codewords. The distance between the codewords is  $n$ .  
 $(000 \dots 0) \quad (111 \dots 1)$
- Assuming  $n$  is odd the error correcting capability of the code is  $(n-1)/2$ .
- The number of vectors in the decoding region of a codeword is

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{(n-1)/2}$$

$$\sum_{k=0}^{(n-1)/2} \binom{n}{k} = \boxed{2^{n-1}}$$

•  $c_0$       •  $c_1$

- The total number of vectors in all the decoding regions is

$$2 \times \boxed{2^{n-1}} = \boxed{2^n}$$

- Thus the vectors in all the decoding regions fill up the entire space.



# Geometry of Hamming Code

- Code 4 (the Hamming code) has 16 codewords of length 7. The distance between the codewords is 3.
- The error correcting capability of the code is 1.
- The number of vectors in the decoding region of a codeword is

$$H \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 8$$

$$\sum_{k=0}^1 \binom{7}{k} = 8.$$

- The total number of vectors in all the decoding regions is

$$16 \times 8 = 2^7.$$

# of binary vectors of length 7

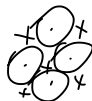
- Thus the vectors in all the decoding regions fill up the entire space.



# Geometry of Code 3

 $\dot{C}_0$  $\dot{C}_1$ 

- Code 3 has 4 codewords of length 9. The distance between the codewords is 6.
- The error correcting capability of the code is 2.
- Consider a decoder that only tries to correct 2 or fewer errors.
- The number of vectors in the decoding region of a codeword is



$$\sum_{k=0}^2 \binom{9}{k} = 46.$$

- The total number of vectors in all the decoding regions is

$$4 \times 46 = 184.$$

- The total number of binary vectors of length 9 is  $2^9 = 512$ .
- Thus the vectors within distance 2 of some codeword represents about 28% of all vectors.

# Geometry of Code 5

- Code 5 has 32 codewords of length 15. The distance between the codewords is 7.
- The error correcting capability of the code is 3.
- Consider a decoder that only tries to correct 3 or fewer errors.
- The number of vectors in the decoding region of a codeword is

$$\sum_{k=0}^3 \binom{15}{k} = 576.$$

- The total number of vectors in all the decoding regions is

$$32 \times 576 = 18432.$$

- The total number of binary vectors of length 15 is  $2^{15} = 32768$ .
- Thus the vectors within distance 3 of some codeword represents about 56.25% of all vectors.

# Bounds on the size of codes

- We can use the geometry of codes to determine bounds on how many codewords there can be while maintaining a certain distance between distinct codewords.
- If  $M$  is the number of codewords of length  $n$  and  $d_{min}$  is the minimum distance then the code can correct  $e$  errors provided  $2e + 1 \leq d_{min}$ .
- The number of vectors in each decoding region will be

$$1 + \binom{n}{1} + \binom{n}{2} + \cdots \binom{n}{e}$$

The total number of vectors in all decoding regions is then

$$M \left[ 1 + \binom{n}{1} + \binom{n}{2} + \cdots \binom{n}{e} \right]$$

- This number must be no greater than the total number of vectors of length  $n$ . So

$$M \left[ 1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{e} \right] \leq 2^n$$

$$M \leq \frac{2^n}{\sum_{m=0}^e \binom{n}{m}}$$

$$r = \frac{\log_2(M)}{n} \leq 1 - \frac{1}{n} \log_2 \left( \sum_{m=0}^e \binom{n}{m} \right)$$

- This is known as the Hamming bound. It gives an upper bound on the rate as a function of the error correcting capability.
- The larger the error correcting capability  $e$ , the smaller the rate of the code.

- Consider the sum of binomial coefficients and parameter  $n$  and suppose that  $p < 1/2$ . Then we can find an approximation to the distribution as follows.

$$\begin{aligned}
 \frac{\sum_{j=0}^{pn} \binom{n}{j}}{p^{-pn}(1-p)^{-(1-p)n}} &= \sum_{j=0}^{pn} \binom{n}{j} (1-p)^{pn} \left(\frac{p}{1-p}\right)^{pn} \\
 &\leq \sum_{j=0}^{pn} \binom{n}{j} (1-p)^{pn} \left(\frac{p}{1-p}\right)^j \\
 &\leq \sum_{j=0}^n \binom{n}{j} (1-p)^{pn} \left(\frac{p}{1-p}\right)^j \\
 &= \sum_{j=0}^n \binom{n}{j} p^j (1-p)^{n-j} = 1
 \end{aligned}$$

- The first inequality is due to the fact that  $p/(1-p) < 1$ . The second inequality is due to the fact that adding more positive terms to the sum increases the sum.

So

$$\sum_{j=0}^{pn} \binom{n}{j} \leq p^{pn} (1-p)^{(1-p)n}$$

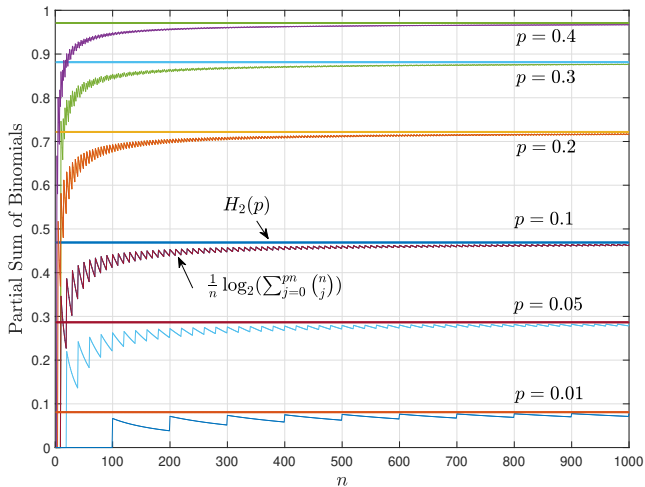
$$\frac{1}{n} \log_2 \left( \sum_{j=0}^{pn} \binom{n}{j} \right) \leq -p \log_2(1-p) - (1-p) \log_2(1-p)$$

$$= H_2(p)$$

where  $H_2(p)$  is the binary entropy function.

$$H_2(p) = -p \log_2 p$$

# $\frac{1}{n} \log_2(\sum_{k=0}^{pn} \binom{n}{k})$ vs. $H_2(p)$



## Lower Bound

- Now consider the best possible set of  $M$  codewords of length  $n$ .
- Suppose the distance for this best possible code is  $d_{min}$  with error correcting capability  $e$  so that  $2e + 1 \leq d_{min}$ .
- Then it must be the the regions around each codeword that are distance  $d_{min}$ .
- Then the set of all such regions should include all possible vectors.





- If there was a vector that was not in one of these regions then that vector could be added to the code. But we started with the best possible set of  $M$  codewords of length  $n$ . So

$$M \sum_{m=0}^{d_{\min}} \binom{n}{m} \geq 2^n$$

$$M \geq \frac{2^n}{\sum_{m=0}^{d_{\min}} \binom{n}{m}}$$

$$r = \frac{\log_2(M)}{n} \geq 1 - \frac{1}{n} \log_2 \left( \sum_{m=0}^{d_{\min}} \binom{n}{m} \right)$$

- In other words, the rate of the best code is at least as big as the right hand of the above equation. This is known as the Gilbert bound.

# Asymptotic Bounds on Distance and Rate of Codes

Let  $n, M, d \rightarrow \infty$  such that  $\frac{\log_2 M}{n} \rightarrow R$  and  $\frac{d_{\min}}{n} \rightarrow \delta$  then the above bound can be written as:

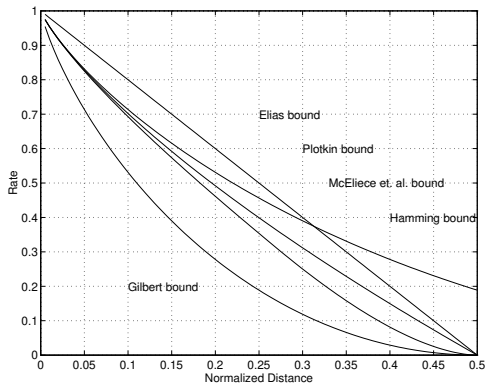
- ① Hamming bound:  $R \leq 1 - H_2(\delta/2)$
- ② Gilbert bound:  $R \geq 1 - H_2(\delta)$
- ③ Singleton bound:  $R \leq 1 - \delta$
- ④ Plotkin:  $R \leq 1 - 2\delta$  ( $0 \leq \delta \leq 1/2$ )
- ⑤ Elias:  $R \leq 1 - H_2(\omega_0)$ ,  $\omega_0 = \frac{1}{2} [1 - \sqrt{1 - 2\delta}]$
- ⑥ McEliece et. al. bound

$$R \leq \min_{0 \leq u \leq 1-2\delta} \{1 + h(u^2) - h(u^2 + 2\delta u + 2\delta)\}$$

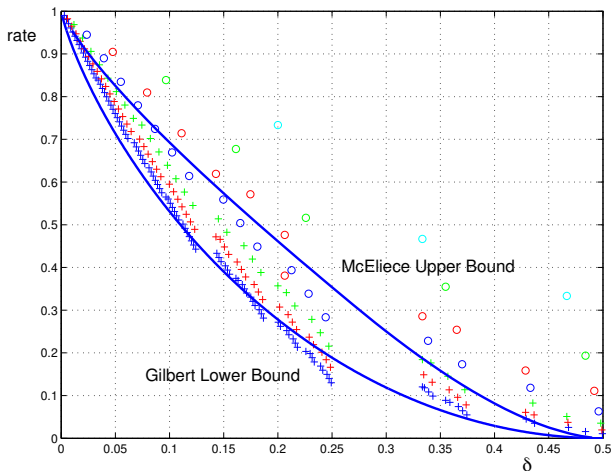
where  $h(x) = H_2(\frac{1}{2}(1 - \sqrt{1-x}))$  and  
 $H_2(p) = -p \log_2(p) - (1-p) \log_2(1-p)$ .

# Bounds on Distance and Rate of Codes

The graphs of these bounds are shown below. The Gilbert bound guarantees existence of codes on or above the lower line while the other bounds are upper limits on the rate of a code for a given distance.



# Bounds on Distance and Rate of Codes



Lower points (+) in blue are length 1023 BCH codes. Lower points (+) in red are length 511 BCH codes. Points (+) in green are length 255 BCH codes. Points (o) in blue are length 127 BCH codes. Points (o) in red are length 63 BCH codes. Points (o) in green are length 31 BCH codes. Points (o) in cyan are length 31 BCH codes

# Bounds on Performance

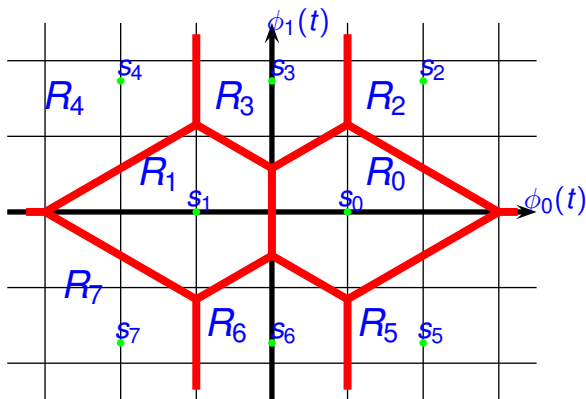
The performance of the maximum likelihood decoder is usually very difficult to evaluate exactly. Usually upper bounds on the decoding error probability are employed (for maximum likelihood decoding). One bound is called the union bound.

Let  $R_i$  be the region of received signals where it is decided that signal  $i$  is transmitted. Let  $R_{i,j}$  be the region where signal  $j$  is chosen when compared only to signal  $i$ . Then

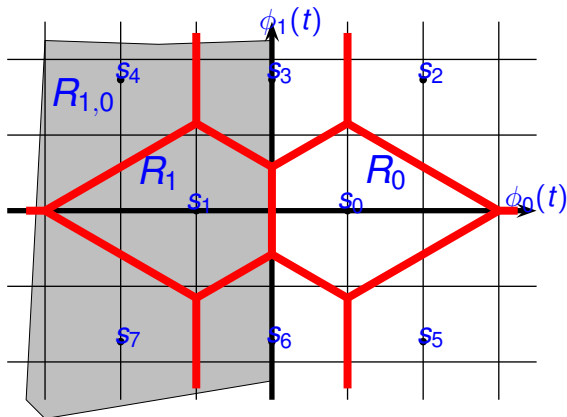
$$R_1 \cup R_2 \cup \cdots \cup R_{M-1} = R_{1,0} \cup R_{2,0} \cdots \cup R_{M-1,0}$$

$$\begin{aligned} P_{e,0} &= P\{\text{error} | s_0 \text{ transmitted}\} \\ &= P\{r \in R_1 \cup R_2 \cup R_3 \cup R_{M-1} | s_0 \text{ transmitted}\} \\ &= P\{r \in R_{0,1} \cup R_{0,2} \cup R_{0,3} \cup R_{0,M-1} | s_0 \text{ transmitted}\} \\ &\leq \sum_{i=1}^{M-1} P\{r \in R_{0,i} | s_0 \text{ transmitted}\} \end{aligned}$$

# Example

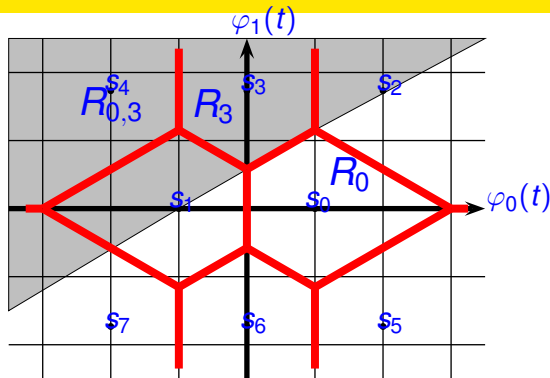


# Example



$$P_2(s_0 \rightarrow s_1) = Q\left(\frac{1}{\sigma}\right) = Q\left(\sqrt{\frac{2}{N_0}}\right)$$

# Example



$$P_2(s_0 \rightarrow s_3) = Q\left(\frac{d_{0,3}}{2\sigma}\right) = Q\left(\frac{2}{2\sigma}\right) = Q\left(\sqrt{\frac{4}{4\sigma^2}}\right) = Q\left(\sqrt{\frac{2}{N_0}}\right)$$

$$P_2(s_0 \rightarrow s_4) = Q\left(\frac{d_{0,4}}{2\sigma}\right) = Q\left(\frac{\sqrt{12}}{2\sigma}\right) = Q\left(\sqrt{\frac{12}{4\sigma^2}}\right) = Q\left(\sqrt{\frac{6}{N_0}}\right)$$



# Pairwise Distance

	$s_0$	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$
$s_0$	0	2	2	2	$2\sqrt{3}$	2	2	$2\sqrt{3}$
$s_1$	2	0	$2\sqrt{3}$	2	2	$2\sqrt{3}$	2	2
$s_2$	2	$2\sqrt{3}$	0	2	4	$2\sqrt{3}$	4	$2\sqrt{7}$
$s_3$	2	2	2	0	2	4	$2\sqrt{3}$	4
$s_4$	$2\sqrt{3}$	2	4	2	0	$2\sqrt{7}$	4	$2\sqrt{3}$
$s_5$	2	$2\sqrt{3}$	$2\sqrt{3}$	4	$2\sqrt{7}$	0	2	4
$s_6$	2	2	4	$2\sqrt{3}$	4	2	0	2
$s_7$	$2\sqrt{3}$	2	$2\sqrt{7}$	4	$2\sqrt{3}$	4	2	0

# Union Bounds

$$\begin{aligned}
 P_{e,0} &= P_{e,1} \\
 &\leq Q\left(\frac{2}{2\sigma}\right) + Q\left(\frac{2}{2\sigma}\right) + Q\left(\frac{2}{2\sigma}\right) + Q\left(\frac{2\sqrt{3}}{2\sigma}\right) + Q\left(\frac{2}{2\sigma}\right) + Q\left(\frac{2}{2\sigma}\right) + Q\left(\frac{2}{2\sigma}\right) \\
 &= 5Q\left(\frac{2}{2\sigma}\right) + 2Q\left(\frac{2\sqrt{3}}{2\sigma}\right) \\
 P_{e,2} &= P_{e,4} = P_{e,5} = P_{e,7} \\
 &\leq 2Q\left(\frac{2}{2\sigma}\right) + 2Q\left(\frac{2\sqrt{3}}{2\sigma}\right) + 2Q\left(\frac{4}{2\sigma}\right) + Q\left(\frac{2\sqrt{7}}{2\sigma}\right) \\
 P_{e,3} &= P_{e,6} \\
 &\leq 4Q\left(\frac{2}{2\sigma}\right) + Q\left(\frac{2\sqrt{3}}{2\sigma}\right) + 2Q\left(\frac{4}{2\sigma}\right)
 \end{aligned}$$

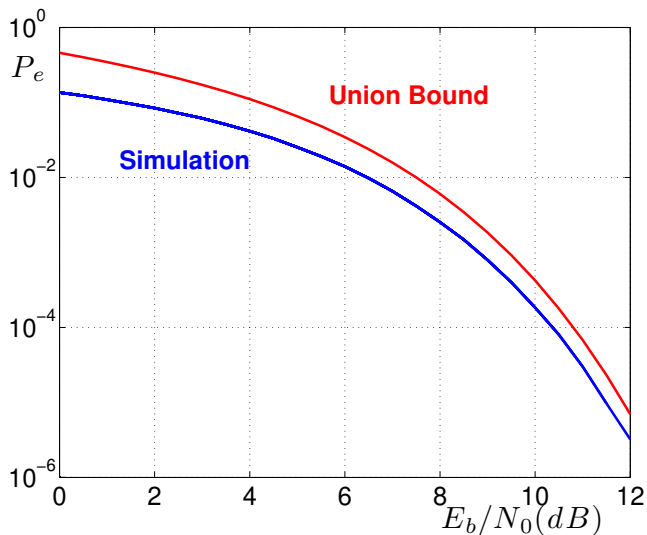
# Union Bounds

$$\begin{aligned}
 P_e &= \frac{1}{8} \sum_{i=0}^7 P_{e,i} \\
 &\leq \frac{1}{8} \left[ 26Q\left(\frac{2}{2\sigma}\right) + 14Q\left(\frac{2\sqrt{3}}{2\sigma}\right) + 12Q\left(\frac{4}{2\sigma}\right) + 4Q\left(\frac{2\sqrt{7}}{2\sigma}\right) \right]
 \end{aligned}$$

~ 0 0 0 0 0 0 0 0  
 0 0 0 0 0 1 1

$b = 2$   
 $s = 1$

# Union Bounds



# Union Bounds

The union bound on the probability of error given signal  $s_i$  transmitted is given by

$$P_{e,i} \leq \sum_{d=d_{\min}}^n A_d P_2(d)$$

where  $P_2(d)$  is the error probability between two codewords of a repetition code of length  $d$  and  $A_d$  is the number of codewords of distance  $d$  from codeword  $i$ . The sum extends from  $d = d_{\min}$  the minimum nonzero weight, to  $n$  the length of the code. The bound is called the union bound. Notice that the effect on performance of the code is completely determined by  $A_d$  while the effect of the channel on the performance is determined by  $P_2(d)$ . For linear codes (described below) the codeword error probability does not depend on  $i$ .

different channel.

# Pairwise Error Probability

For an additive white Gaussian noise channel with BPSK modulation the pairwise error probability between two codewords that differ in  $d$  positions is

$$P_2(d) = Q\left(\frac{dE}{2\sigma}\right) = Q\left(\sqrt{\frac{2Ed_H}{N_0}}\right).$$

$$\begin{aligned} s_0 &= (+1E, +1E, \dots, +1E) \\ s_1 &= (-1E, -1E, \dots, -1E) \end{aligned}$$

Note that  $E$  is the energy per code symbol. The energy per bit is

$$E_b = nE / \log_2(M) = E/r$$

where  $r = \log_2(M)/n$  is the rate of the code. Thus

$$P_2(d) = Q\left(\sqrt{\frac{2E_b r d_H}{N_0}}\right).$$

$$\begin{aligned} d_E^2(s_0, s_1) &= 4E d_H \\ \frac{d_E^2(s_0, s_1)}{2\sqrt{N_0}} &= \frac{\sqrt{4E d_H}}{2\sqrt{N_0}} \\ &= \sqrt{\frac{2E d_H}{N_0}} \end{aligned}$$

Thus a key parameter of a code is the product of the rate and distance.

# Pairwise Error Probability

BSC channel

different  
channel

For a binary symmetric channel the pairwise error probability between two codewords that differ in  $d$  positions is

$$P_2(d) = \begin{cases} \sum_{l=e+1}^d \binom{d}{l} p^l (1-p)^{d-l} & d \text{ odd} \\ \sum_{l=e+2}^d \binom{d}{l} p^l (1-p)^{d-l} + \frac{1}{2} \binom{d}{d/2} p^{d/2} (1-p)^{d/2} & d \text{ even} \end{cases}$$

where  $e = \lfloor (d-1)/2 \rfloor$ . The  $d$  even expression accounts for the possibilities of ties.

# Union-Bhattacharyya Bound

This bound can be simplified greatly (and loosened) for binary codes by employing the Bhattacharyya bound. The result is

$$P_e \leq \sum_{d=d_{\min}}^n A_d D^d$$

where

$$D = \sum_z \sqrt{p(z|0)p(z|1)}$$

Again the effect on performance of the code is through the weights  $A_d$  while the effect of the channel is through the parameter  $D$ .

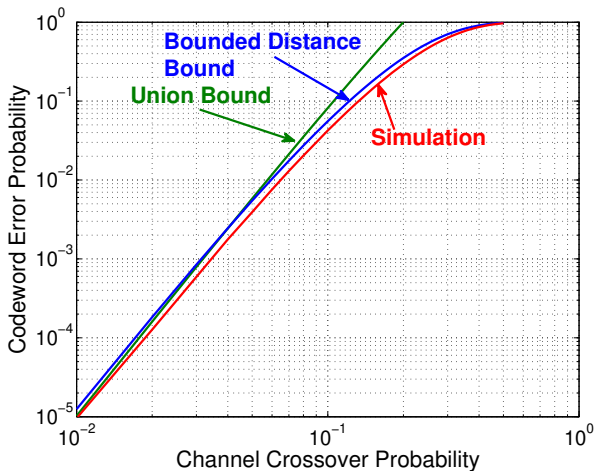


# Union Bound on (15,5) BCH code

$$n=32$$

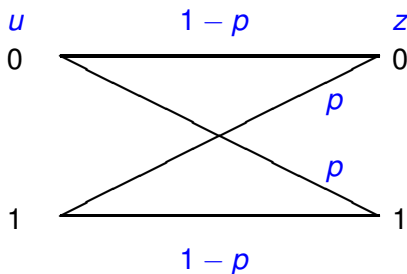
$$d_{H,min}=7$$

$$e=3$$



# Union-Bhattacharyya Bound

For a binary symmetric channel



$$\begin{aligned}
 D &= \sqrt{p(1|0)(p(0|0))} + \sqrt{p(0|1)p(1|1)} \\
 &= \sqrt{p(1-p)} + \sqrt{(1-p)p} \\
 &= 2\sqrt{p(1-p)}
 \end{aligned}$$

# Union-Bhattacharyya Bound

For an additive white Gaussian noise channel

X

$$\begin{aligned}
 D &= \int_z \sqrt{p(z|0)p(z|1)} dz \\
 &= \int_z \left[ \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\sqrt{E})^2/(2\sigma^2)} \frac{1}{\sqrt{2\pi}\sigma} e^{-(z+\sqrt{E})^2/(2\sigma^2)} \right]^{1/2} dz \\
 &= e^{-E/N_0}
 \end{aligned}$$

# Repetition Code on Binary Symmetric Channel.

$d$	$A_d$
0	1
1	0
2	0
3	0
4	0
5	1

X

# Repetition Code on Binary Symmetric Channel.

## Union Bound

$$C_0 = (000 \dots 0)$$

$$C_1 = (111 \dots 1)$$

$$P_e \leq \begin{cases} \sum_{l=t}^n \binom{n}{l} p^l (1-p)^{n-l} & n \text{ odd} \\ \sum_{l=t+1}^n \binom{n}{l} p^l (1-p)^{n-l} + \frac{1}{2} \binom{n}{n/2} p^{n/2} (1-p)^{n/2} & n \text{ even} \end{cases}$$

$$t = \lfloor (n+1)/2 \rfloor$$

## Union-Bhattacharyya Bound

$$P_e \leq 1D^n = (2\sqrt{p(1-p)})^n$$

$$D = 2\sqrt{p(1-p)}$$

Note: When there are only two codewords the union bound is actually also the actual error probability.

# Repetition Code on AWGN Channel.

## Union Bound

$E_b = E$  here

$$\begin{aligned}
 P_e &\leq Q\left(\frac{d_E}{2\sqrt{N_0/2}}\right) = Q\left(\sqrt{\frac{4nE}{2N_0}}\right) \\
 &= Q\left(\sqrt{\frac{(2nE)}{N_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)
 \end{aligned}$$

## Union-Bhattacharyya Bound

$$\begin{aligned}
 P_e &\leq D^n = e^{-En/N_0} = e^{-E_b/N_0} \\
 D &= e^{-E/N_0}
 \end{aligned}$$

Note: When there are only two codewords the union bound is actually also the actual error probability.

# Code 2 on an AWGN Channel

0000  
 0011  
 1100  
 1111

$d$	$A_d$
0	1
1	0
2	2
3	0
4	1

# Code 2 on an AWGN Channel

$\geq$  *codeword with  $d=2$*       Union Bound      *1 codeword with  $d=4$*

$$P_e \leq 2Q\left(\sqrt{\frac{2E}{N_0}}\right) + Q\left(\sqrt{\frac{2(4E)}{N_0}}\right)$$

$$= 2Q\left(\sqrt{\frac{2(E_b(1/2)2)}{N_0}}\right) + Q\left(\sqrt{\frac{2(E_b(1/2)4)}{N_0}}\right)$$

Union-Bhattacharyya Bound

$$\begin{aligned}
 D &= e^{-E/N_0} \\
 P_e &\leq 2D^2 + D^4 \\
 &= 2e^{-2E/N_0} + e^{-4E/N_0} \\
 &= 2e^{-E_b/N_0} + e^{-2E_b/N_0}
 \end{aligned}$$



# Hamming code on a binary symmetric channel

## Union Bound

$$P_e \leq 7P_2(3) + 7P_2(4) + P_2(7)$$

where

$$P_2(3) = \sum_{l=2}^3 \binom{3}{l} p^l (1-p)^{3-l}$$

$$P_2(4) = \sum_{l=3}^4 \binom{4}{l} p^l (1-p)^{4-l} + \frac{1}{2} \binom{4}{2} p^2 (1-p)^2$$

$$P_2(7) = \sum_{l=4}^7 \binom{7}{l} p^l (1-p)^{7-l}$$



7 code words

distance 3, 4

1 code word  
distance 7

# Hamming code on a binary symmetric channel

## Union-Bhattacharyya Bound

$$\begin{aligned} P_e &\leq 7D^3 + 7D^4 + D^7 \\ &= 7[2\sqrt{p(1-p)}]^3 + 7[2\sqrt{p(1-p)}]^4 + [2\sqrt{p(1-p)}]^7 \end{aligned}$$

# Hamming Code on an AWGN Channel

$$\text{rate} = \frac{4}{7}$$

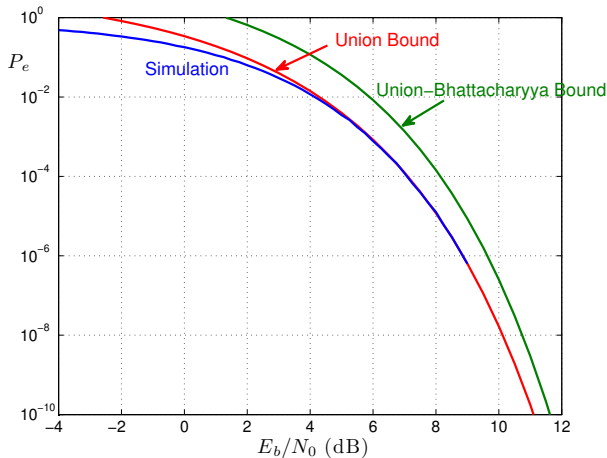
Union Bound

$$P_e \leq 7Q\left(\sqrt{\frac{2E_b(4/7)3}{N_0}}\right) + 7Q\left(\sqrt{\frac{2E_b(4/7)4}{N_0}}\right) + 1Q\left(\sqrt{\frac{2E_b(4/7)7}{N_0}}\right)$$

Union-Bhattacharyya Bound

$$\begin{aligned} P_e &\leq 7D^3 + 7D^4 + D^7 \\ &= 7e^{-[E_b(4/7)3/N_0]} + 7e^{-[E_b(4/7)4/N_0]} + e^{-[E_b(4/7)7/N_0]} \end{aligned}$$

# Codeword Error Probability for Hamming Code on an AWGN Channel



# Code 5 on an AWGN Channel

## Union Bound

$$\begin{aligned}
 P_e &\leq 15Q\left(\sqrt{\frac{2(7E)}{N_0}}\right) + 15Q\left(\sqrt{\frac{2(8E)}{N_0}}\right) + 1Q\left(\sqrt{\frac{2(15E)}{N_0}}\right) \\
 &= 15Q\left(\sqrt{\frac{2(E_b(1/3)7)}{N_0}}\right) + 15Q\left(\sqrt{\frac{2(E_b(1/3)8)}{N_0}}\right) + Q\left(\sqrt{\frac{2(E_b(1/3)15)}{N_0}}\right)
 \end{aligned}$$

## Union-Bhattacharyya Bound

$$\begin{aligned}
 D &= e^{-E/N_0} \\
 P_e &\leq 15D^7 + 15D^8 + 1D^{15} \\
 &= 15e^{-7E/N_0} + 15e^{-4E/N_0} \\
 &= 15e^{-(7/3)E_b/N_0} + 15e^{-8/3E_b/N_0} + 1e^{-15/3E_b/N_0}
 \end{aligned}$$

# Complexity of Decoding

- For an arbitrary code the complexity of maximum likelihood decoding is large if  $M$  is large.
- Typical values of  $M$  are
  - A CD player has  $M = 2.7 \times 10^{67}$  for one of the short Reed-Solomon codes.
  - The NASA standard (255,223) code has  $M = 256^{223} = 1.1 \times 10^{537}$ .
  - The shortest code used in WiFi has parameters  $n = 648$  and  $M = 2^{324} = 3.4 \times 10^{97}$ .
- Clearly this does not give a practical implementation.
- Thus we are forced to impose some structure on the code or to use suboptimum decoding (or both) to reduce the complexity of decoding.
- The structure imposed is linearity and cyclicity.