ECE 5654 Lecture 14
Error Correcting Codes

Ravi Tandon

Virginia Tech

March 25, 2014

Learning Objectives

At the end of this lecture, the student should be able to:

- Define Linear Block Codes, state what is a generator matrix, parity check matrix and how they are used in encoding/decoding
- Describe the key parameters of any error correcting code, i.e., error correcting, error detecting capabilities
- Decoding linear codes; performance of block codes

Block Codes

- Let $\mathbf{m} = (m_1, m_2, \ldots, m_k)$ be a block of k message bits
- There are 2^k distinct possible message blocks
- (n, k) block code produces an n-bit codeword for each message block
- Let $\mathbf{c} = (c_1, c_2, \ldots, c_n)$ denote a n length codeword
- Hence, a code is a one-to-one mapping from
  2^k message blocks (each block of length k)
  to
  2^k codewords (each codeword of length n)
- We denote a code by $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_{2^k}\}$

$(n, k) = (7, 4)$ Hamming Code

Input	Codeword	Weight
0000	0000000	0
0001	0001110	3
0010	0010101	3
0011	0011011	4
0100	0100011	3
0101	0101101	4
0110	0110110	4
0111	0111000	3
1000	1000111	4
1001	1001001	3
1010	1010010	3
1011	1011100	4
1100	1100100	3
1101	1101010	4
1110	1110001	4
1111	1111111	7

- Total of $2^k = 2^4 = 16$ possible message blocks
- For each block, we have a corresponding $n = 7$ length codeword
- Code: $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{16}\}$

Linear Block Codes

- A code $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_{2^k}\}$ is said to be linear if any two code words in the code can be added to produce a third code word in the code.
- For a linear code: if $\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}$, then $\mathbf{c}_1 \oplus \mathbf{c}_2 \in \mathcal{C}$
- A code $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_{2^k}\}$ is said to be cyclic if $\mathbf{c} = (c_1, c_2, \ldots, c_n) \in \mathcal{C}$, then $(c_n, c_1, \ldots, c_{n-1}) \in \mathcal{C}$
- Most practical block codes are linear and cyclic
- We will focus on linear block codes
- $(7, 4)$ Hamming code is linear and cyclic

Linear Block Codes

- If the first k bits of every codeword are the same as the k message bits, then the code is said to be systematic.
- The last $(n-k)$ bits are also known as parity check bits or parity bits
- These $(n-k)$ parity bits are linear sums of the k message bits

$$b_i = p_{1i}m_1 + p_{2i}m_2 + \ldots + p_{k,i}m_k$$

- In a vector form:

$$\begin{aligned}\mathbf{m} &= [m_1, m_2, \ldots, m_k] \ (k \text{ message bits}) \\ \mathbf{b} &= [b_1, b_2, \ldots, b_{n-k}] \ ((n-k) \text{ parity bits}) \\ \mathbf{c} &= [\mathbf{m} : \mathbf{b}] \ (n \text{ length codeword})\end{aligned}$$

- $(7, 4)$ hamming code is systematic

Generator Matrix

- We define the $k \times (n-k)$ coefficient matrix P as

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1,n-k} \\ p_{21} & p_{22} & \cdots & p_{2,n-k} \\ \vdots & \vdots & \vdots & \vdots \\ p_{k,1} & p_{k,2} & \cdots & p_{k,n-k} \end{bmatrix}$$

- For message block, the corresponding parity bits can be obtained as

$$\underbrace{\mathbf{b}}_{1\times(n-k)} = \underbrace{\mathbf{m}}_{1\times k} \underbrace{\mathbf{P}}_{k\times(n-k)}$$

- Mapping from k-length message to n-length codeword:

$$\mathbf{c} = [\mathbf{m} : \mathbf{b}] = [\mathbf{m} : \mathbf{mP}] = \mathbf{m}[\mathbf{I}_k : \mathbf{P}] = \mathbf{mG}$$

- $\mathbf{G} = [\mathbf{I}_k : \mathbf{P}]$ is called as the Generator matrix of the code

Generator Matrix

- Generator matrix: $\mathbf{G} = [\mathbf{I}_k : \mathbf{P}]$ is a $k \times n$ matrix
- Matrix $\mathbf{P}$ is selected such that rows of $\mathbf{G}$ are linearly independent
- The sum of two codewords is another codeword:

$$\mathbf{c}_i \oplus \mathbf{c}_j = \mathbf{m}_i\mathbf{G} \oplus \mathbf{m}_j\mathbf{G} = (\mathbf{m}_i \oplus \mathbf{m}_j)\mathbf{G}$$

- There is another way to represent the relationship between message bits and parity check bits
- Define the $(n-k) \times n$ Parity-check matrix:

$$\mathbf{H} = [-\mathbf{P}^T : \mathbf{I}_{n-k}]$$

- It then follows that:

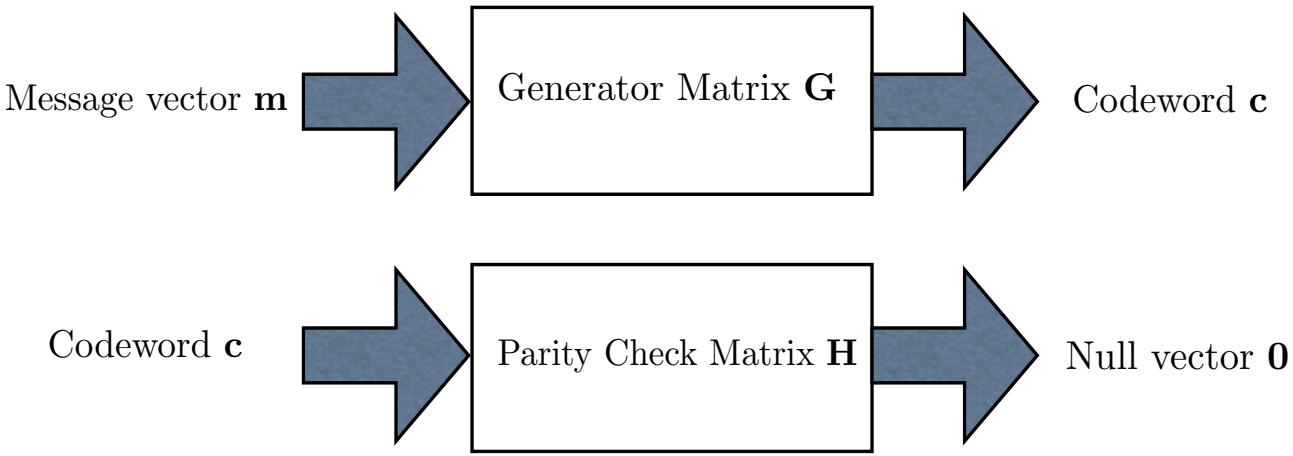
$$\mathbf{G}\mathbf{H}^T = [\mathbf{I}_k : \mathbf{P}] \begin{bmatrix} -\mathbf{P} \\ \mathbf{I}_{n-k} \end{bmatrix} = \mathbf{0}$$

Parity Check Matrix

- Generator and parity check matrices satisfy: $\mathbf{GH}^T = \mathbf{0}$
- For any codeword $\mathbf{c}$, we can also show (using the above property)

$$\mathbf{cH}^T = \mathbf{mGH}^T = \mathbf{0}$$

- This property is extremely useful and is exploited in error correction, error detection.



Example: $(n, k) = (7, 4)$ Hamming Code

- Generator matrix: $[I_4 : P]$

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

- Parity Check matrix: $[P^T : I_3]$

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

- If $\mathbf{m} = (m_1, m_2, m_3, m_4)$ is information sequence, then codeword is

$$c_1 = m_1, c_2 = m_2, c_3 = m_3, c_4 = m_4,$$
$$c_5 = m_1 + m_2 + m_3$$
$$c_6 = m_2 + m_3 + m_4$$
$$c_7 = m_1 + m_2 + m_4$$

Minimum Distance

- Minimum distance of a code:

$$d_{\min} = \min_{c_1, c_2 \in \mathcal{C}} d(c_1, c_2)$$

- Minimum weight of a code:

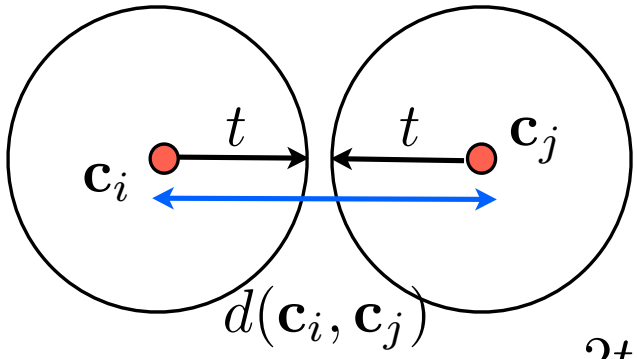
$$w_{\min} = \min_{c \neq 0} w(c) = d(c, 0)$$

- Minimum Distance (of a linear block code)

$$d_{\min} = w_{\min}$$

- That is, minimum distance of a linear block code is the minimum weight over all non-zero codewords.

Error Correction and Error Detection Capability



$$2t < d(\mathbf{c}_i, \mathbf{c}_j) \quad \text{for all} \quad i, j$$

$$2t + 1 \leq d(\mathbf{c}_i, \mathbf{c}_j) \quad \text{for all} \quad i, j$$

$$2t + 1 \leq \min_{i,j} d(\mathbf{c}_i, \mathbf{c}_j)$$

$$2t + 1 \leq d_{\min} \Rightarrow t \leq \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$$

- Any code with minimum distance $d_{\min}$ can correct any combination of up to $t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$ errors
- We call t as the error correcting capability of the code
- Any code with minimum distance $d_{\min}$ can detect any combination of up to $d_{\min} - 1$ errors
- Logic: if we send a codeword $\mathbf{c}$, and the receiver gets $\mathbf{r}$, it can compute the distance $d(\mathbf{c}, \mathbf{r})$. If $d(\mathbf{c}, \mathbf{r}) < d_{\min}$, then either $\mathbf{r}$ is either equal to $\mathbf{c}$ or $\mathbf{r}$ is not a codeword. Hence, as long as the number of errors is $< d_{\min}$, the receiver can detect that an error occured.

Minimum Distance

- It is clear that it is desirable to have a large value of $d_{\min}$
- How large can $d_{\min}$ be ?
- Singleton bound shows that $d_{\min} \leq n - k + 1$
- Logic behind this bound:
  - Let C be an arbitrary block code of minimum distance $d_{\min}$
  - All codewords of this code are distinct; and there are 2^k codewords
  - Suppose that we delete the first $(d_{\min} - 1)$ letters of each codeword
  - The resulting codewords must still be pairwise different (since original codewords had a minimum distance of $d_{\min}$)
  - So, we are left with codewords of length $n - (d_{\min} - 1)$ and there can be $2^{n-(d_{\min}-1)}$ possible sequences of this length
  - Hence, $2^k \leq 2^{n-(d_{\min}-1)}$, which implies $k \leq n - (d_{\min} - 1)$ or $d_{\min} \leq n - k + 1$
- Codes with minimum distance $d_{\min} = n - k + 1$ are called as Maximum Distance Separable (MDS) codes

Decoding of Block Codes

- Syndrome Decoding
  - only useful for short codes
- Bounded Distance Decoding
  - Berlekamp-Massey Algorithm
  - Can be used on a large class of linear, cyclic codes
  - Widely used in decoding of BCH, Reed-Solomon codes
  - We will not cover this in this course

Syndrome Decoding of Block Codes

- Let $\mathbf{r} = \mathbf{c} + \mathbf{e}$ be the vector of n received bits
- $\mathbf{e}$ is sometimes called the error vector (or error pattern)
- Syndrome decoding can be thought of as a method of realizing minimum-distance decoding
- Then, $\mathbf{s}$ is the syndrome of the received vector, where

$$\mathbf{s} = \mathbf{r}\mathbf{H}^T = (\mathbf{c} + \mathbf{e})\mathbf{H}^T = \mathbf{c}\mathbf{H}^T + \mathbf{e}\mathbf{H}^T = \mathbf{e}\mathbf{H}^T$$

- Syndrome $(1 \times (n - k))$ depends only on the error vector $\mathbf{e}$
- Total number of possible error patterns: $2^n - 1$
  (we do not count all zero as an error)
- Total number of possible syndromes: 2^{n-k}
- Multiple error patterns can give the same syndrome
  (and all error patterns are not correctable)

Syndrome Decoding (continued)

$$\begin{array}{ccccc} c_1 = \mathbf{0} & c_2 & c_3 & \cdots & c_{2^k} \\ e_2 & c_2 + e_2 & c_3 + e_2 & \cdots & c_{2^k} + e_2 \\ e_3 & c_2 + e_3 & c_3 + e_3 & \cdots & c_{2^k} + e_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ e_{2^{n-k}} & c_2 + e_{2^{n-k}} & c_3 + e_{2^{n-k}} & \cdots & c_{2^k} + e_{2^{n-k}} \end{array}$$

- Standard Array consists of 2^{n-k} rows
- Each row is called a coset
- First term (an error pattern) is the coset leader
- Syndrome decoding
  - Suppose that the error pattern is one of the coset leaders (say e_i)
  - Receive $y = c_m + e_i$
  - Find the syndrome of y: $s = yH^T = (c_m + e_i)H^T = e_iH^T$
  - Among all coset leaders with syndrome s, find e_i with smallest weight
  - Add the coset leader with y to obtain the most likely codeword

Example: Syndrome Decoding

- $(n, k) = (5, 2)$ code with the following generator matrix

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

- Codewords: $\{00000, 01011, 10101, 11110\}$
- $d_{\text{min}} = 3$; error correction capability $= 1$
- Standard array:

00000	**01011**	**10101**	**11110**
00001	01010	10100	11111
00010	01001	10111	11100
00100	01111	10001	11010
01000	00011	11101	10110
10000	11011	00101	01110
11000	10011	01101	00110
10010	11001	00111	01100

Probability of Codeword Error

- We wish to compute the probability P_C (probability of codeword error) that the decoder will fail
- The decoder can correct up to, but not more than $t = \lfloor (d_{\text{min}} - 1)/2 \rfloor$ errors
- We assume that the probability of an individual bit error is p, and that bit errors occur independently
- Note: the bit error probability p is determined from the modulation type

Probability of Codeword Error

- If we send n bits, the probability of receiving a specific pattern of i errors and $(n-i)$ correct bits is $p^i(1-p)^{n-i}$
- There are a total of $\binom{n}{i}$ distinct patterns of n bits with i errors and $(n-i)$ correct bits
- Hence, the total probability of receiving a pattern with i errors is

$$\binom{n}{i} p^i (1-p)^{n-i}$$

- Since we can correct any pattern of up to t errors, the overall probability of codeword error is:

$$P_C = 1 - \sum_{i=0}^{t} \binom{n}{i} p^i (1-p)^{n-i}$$

$$= \sum_{i=t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

Example: $(7, 4)$ Hamming Code

- Assume we are using a $(7, 4)$ Hamming Code
- $d_{\text{min}} = 3$ and hence $t = 1$ (it can correct 1 error)
- Assume $p = 0.001$
- For this example, there are fewer terms so it is easier to compute the summation:

$$
\begin{aligned}
P_C &= 1 - \sum_{i=0}^{t} \binom{n}{i} p^i (1-p)^{n-i} \\
&= 1 - \sum_{i=0}^{1} \binom{7}{i} (0.001)^i (0.999)^{n-i} \\
&= 1 - 1 \times (0.999)^7 - 7 \times (0.001) \times (0.999)^6 \\
&= 2 \times 10^{-5}
\end{aligned}
$$

Numerical Evaluation of P_C

- May need to use higher numerical precision when evaluating

$$P_C = 1 - \sum_{i=0}^{t} \binom{n}{i} p^i (1-p)^{n-i}$$
$$= \sum_{i=t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

- In general, $\binom{n}{i}$ can be very large and $p^i(1-p)^{n-i}$ can be very small.
- Frequently, the term $i = t+1$ and the first few terms thereafter are the most significant

Role of coding and E_b/N_0

- We frequently want to evaluate the performance in terms of E_b/N_0
- When using coding, we are sending extra bits ($(n-k)$ bits of redundant information for every k bits)
- In order to make a fair comparison with uncoded systems, we must penalize by the extra energy used to send those bits
- We replace E_b/N_0 by rE_b/N_0 in all our error formulas for different modulation types, where $r = \frac{k}{n}$

Example

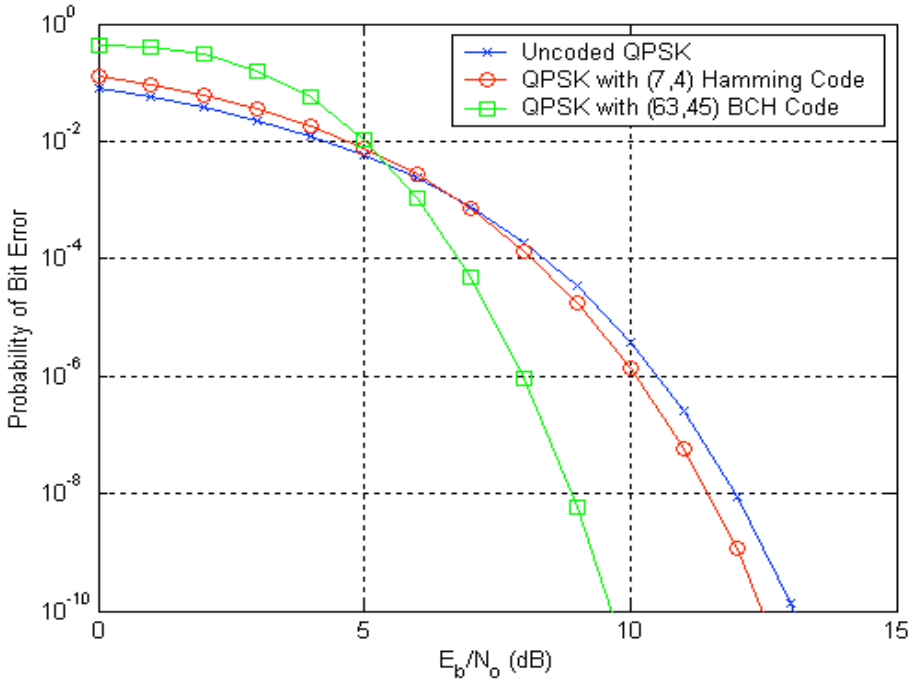
- Suppose we use BPSK modulation and we have $E_b/N_0 = 10dB$.
- We compare the probability of error both for an uncoded system and for a system employing a $(63, 45)$ code with error correction capability of $t = 3$
- Uncoded System: $P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = 3.9e^{-006}$
- Coded System: $p = Q\left(\sqrt{\frac{2\times r\times E_b}{N_0}}\right) = 7.9e^{-005}$
- $r = \frac{45}{63}$
- Probability of codeword error (for a coded system):

$$P_C \approx \sum_{i=4}^{8} \binom{63}{i} (7.9e^{-005})^i (1 - 7.9e^{-005})^{63-i}$$
$$= 2.3e^{-011}$$

Relating Codeword Error Rate and Bit Error Rate

- If the codeword is correctly received, all bits will be correctly received
- If a codeword is incorrectly decoded, a good approximation is that $1/2$ of the bits will be in error
- More exact analytical evaluation of bit error rate is tedious for block codes

Comparison



$(7, 4)$ Hamming Code is very weak

BCH Code much stronger

For smaller E_b/N_0, coding degrades performance

- In the next lecture, we will talk about various classes of block codes, and their applications
- We will also discuss hard-decision vs soft-decision decoding