4F5: Advanced Wireless Communications

Handout 3: Linear Block Codes

Jossy Sayir

Signal Processing and Communications Lab
Department of Engineering
University of Cambridge
jossy.sayir@eng.cam.ac.uk

Lent 2012

Outline

1. Introduction and Motivation
2. Linear Block Codes
3. Error Probability and Union Bound
4. Random Coding for the Binary Erasure Channel

Outline

1 Introduction and Motivation

2 Linear Block Codes

3 Error Probability and Union Bound

4 Random Coding for the Binary Erasure Channel

Outline

1. Introduction and Motivation
2. Linear Block Codes
3. Error Probability and Union Bound
4. Random Coding for the Binary Erasure Channel

Outline

1. Introduction and Motivation
2. Linear Block Codes
3. Error Probability and Union Bound
4. Random Coding for the Binary Erasure Channel

Introduction and Motivation

So far...

- We have shown that we can achieve $P_e \to 0$
  - with codes of rate $R < C$
  - provided that $n \to \infty$
  - using random coding (average P_e over the ensemble of random codes)
- The proof is not constructive
  - average performance
  - does not tell us how to achieve the limits
- Random codes
  - not implementable, we need to store the whole codebook at transmitter and receiver
  - we do not know how to encode and decode algorithmically
  - in practice $n < \infty$, i.e., finite-length codes
- We need codes that can be implemented (encoding and decoding) and that perform close to capacity
- We will study
  - Linear block codes
  - convolutional codes
  - turbo-codes
  - low-density parity-check codes

Linear Block Codes

Definitions

- A binary code $\mathcal{C}$ of length n and dimension k is a set of different 2^k binary codewords of length n.
- The rate of the code is $R = \frac{1}{n}\log_2|\mathcal{C}| = \frac{k}{n}$
- $\mathcal{C}$ is a vector subspace of the vector space defined by all possible binary vectors of length n, hence the code is linear
- $\mathcal{C}$ is the set of codewords $\boldsymbol{c}$ satisfying for all $\boldsymbol{b} \in \mathbb{F}_2^k$ (row convention)

$$\boldsymbol{c} = \boldsymbol{b}\boldsymbol{G}, \quad \text{where} \quad \boldsymbol{G} = \begin{bmatrix} g_{1,1} & \cdots & g_{1,n} \\ g_{2,1} & \cdots & g_{2,n} \\ \vdots & \ddots & \vdots \\ g_{k,1} & \cdots & g_{k,n} \end{bmatrix} \quad \text{is the generator matrix}$$

- For equiprobable messages, every symbol of a linear code is uniformly distributed
- The code is called systematic if the information bits $\boldsymbol{b}$ are part of the codeword, i.e., $\boldsymbol{c} = [\boldsymbol{b}\ \boldsymbol{p}]$ where $\boldsymbol{p} \in \mathbb{F}_2^{n-k}$ is the parity vector (redundancy)
- The corresponding generator matrix is

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{I}_k & \boldsymbol{P} \end{bmatrix}, \quad \text{where} \quad \boldsymbol{P} \in \mathbb{F}_2^{k \times n-k} \quad \text{is the parity generator matrix}$$

Linear Block Codes

Definitions

- We can also express the code $\mathcal{C}$ as the set of codewords $\boldsymbol{c}$ such that

$$\boldsymbol{c}\boldsymbol{H}^T = \boldsymbol{0}, \quad \text{where} \quad \boldsymbol{H} = \begin{bmatrix} h_{1,1} & \dots & h_{1,n} \\ h_{2,1} & \dots & h_{2,n} \\ \vdots & \ddots & \vdots \\ h_{n-k,1} & \dots & h_{n-k,n} \end{bmatrix} \text{ is the parity-check matrix}$$

- $\boldsymbol{H}$ represents the linear system of equations that every codeword must satisfy
- The parity-check matrix of a systematic code can be expressed as

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{P}^T & \boldsymbol{I}_{n-k} \end{bmatrix}$$

- Hamming weight $w_{\mathrm{h}}(\boldsymbol{c}) = \sum_{i=1}^{n} c_i$, sum is the sum over the integers (not binary)
- Hamming distance between $\boldsymbol{c}, \boldsymbol{c}' \in \mathcal{C}$: number of positions in which they differ

$$d_{\mathrm{h}}(\boldsymbol{c}, \boldsymbol{c}') = \sum_{i=1}^{n} c_i \oplus c_i' = w_{\mathrm{h}}(\boldsymbol{c} \oplus \boldsymbol{c}')$$

- Minimum Hamming distance

$$d_{\min} = \min_{\substack{\boldsymbol{c}, \boldsymbol{c}' \in \mathcal{C} \\ \boldsymbol{c}' \neq \boldsymbol{c}}} d_{\mathrm{h}}(\boldsymbol{c}, \boldsymbol{c}')$$

- Since the sum of 2 codewords is a codeword (linear code)

$$d_{\min} = \min_{\substack{\boldsymbol{c} \in \mathcal{C} \\ \boldsymbol{c} \neq \boldsymbol{0}}} w_{\mathrm{h}}(\boldsymbol{c})$$ we can take the all-zero codeword as reference

Linear Block Codes

Definitions

- Weight enumerator A_d is the number of codewords in $\mathcal{C}$ with Hamming weight d
- Input-output weight enumerator $A_{i,d}$ is the number of codewords in $\mathcal{C}$ with Hamming weight d generated with an input sequence $\boldsymbol{b}$ of Hamming weight $i = w_{\mathrm{h}}(\boldsymbol{b})$
- Obviously, $A_d = \sum_i A_{i,d}$

Example (The $(7,4)$ Hamming Code)

- Binary code of rate $R = \frac{4}{7}$, generator and parity-check matrices given by

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad \boldsymbol{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

- $2^k = 16$ codewords

$$\begin{array}{llll} \boldsymbol{c}_1 = [0000000] & \boldsymbol{c}_2 = [0001111] & \boldsymbol{c}_3 = [0010011] & \boldsymbol{c}_4 = [0011101] \\ \boldsymbol{c}_5 = [0100101] & \boldsymbol{c}_6 = [0101010] & \boldsymbol{c}_7 = [0110110] & \boldsymbol{c}_8 = [0111001] \\ \boldsymbol{c}_9 = [1000110] & \boldsymbol{c}_{10} = [1001001] & \boldsymbol{c}_{11} = [1010101] & \boldsymbol{c}_{12} = [1011010] \\ \boldsymbol{c}_{13} = [1100011] & \boldsymbol{c}_{14} = [1101100] & \boldsymbol{c}_{15} = [1110000] & \boldsymbol{c}_{16} = [1111111] \end{array}$$

- $A_3 = 6, A_4 = 8, A_7 = 1,$
- $A_{1,3} = 3, A_{2,3} = 2, A_{3,3} = 1 A_{1,4} = 1, A_{2,4} = 4, A_{3,4} = 3, A_{4,7} = 1$

Error Probability and Union Bound

Error Probability and Union Bound

- BPSK modulation $x_i = 1 - 2\,c_i,\ i = 1, \ldots, n$ ($0 \longrightarrow +1$ and $1 \longrightarrow -1$)
- Binary codeword $\boldsymbol{c}$ vs modulated BPSK codeword $\boldsymbol{x}$
- AWGN channel
  - $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{z}$
  - $z_i \sim \mathcal{N}(0, \sigma^2)$
  - $P_{Y|X}(y_i|x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y_i - x)^2}$
  - $P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-\frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{x}\|^2}$
- Maximum Likelihood decoding

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}\in\mathcal{C}} P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x}) = \arg\max_{\boldsymbol{x}\in\mathcal{C}} e^{-\frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{x}\|^2}$$
$$= \arg\min_{\boldsymbol{x}\in\mathcal{C}} \|\boldsymbol{y} - \boldsymbol{x}\|^2 = \arg\min_{\boldsymbol{x}\in\mathcal{C}} \sum_{i=1}^{n} (y_i - x_i)^2$$

- Exhaustive search over 2^k codewords, find the closest. Implementable for short codes (i.e., Hamming code), impractical for standard code lengths.

Error Probability and Union Bound

Error Probability and Union Bound

- Calculating exact the error probability for a particular code is a hard task
- However, it is easy to obtain a simple and tight bound using the union bound

$$
\begin{aligned}
P_e &= \Pr\{\hat{\mathbf{c}} \neq \mathbf{0} | \mathbf{0} \text{ was transmitted}\} \\
&= \Pr\left\{ \bigcup_{\hat{\mathbf{c}} \neq \mathbf{0}} \{\text{error with codeword } \hat{\mathbf{c}} | \mathbf{0} \text{ was transmitted}\} \right\} \\
&\leq \sum_{\hat{\mathbf{c}} \neq \mathbf{0}} \Pr\{\text{error with codeword } \hat{\mathbf{c}} \big| \mathbf{0} \text{ was transmitted}\} = \sum_{\hat{\mathbf{c}} \neq \mathbf{0}} \mathrm{PEP}(\mathbf{0} \to \hat{\mathbf{c}})
\end{aligned}
$$

- $\mathrm{PEP}(\mathbf{0} \to \hat{\mathbf{c}})$ is the pairwise error probability

$$
\mathrm{PEP}(\mathbf{0} \to \hat{\mathbf{c}}) = \Pr\left\{ \sum_{i=1}^{n} (y_i - \hat{x}_i)^2 < \sum_{i=1}^{n} (y_i - (+1))^2 \right\}
$$

$$
= \Pr\left\{ \sum_{i=1}^{d} (y_i - (-1))^2 < \sum_{i=1}^{d} (y_i - (+1))^2 \right\} = \Pr\left\{ \sum_{i=1}^{d} 4y_i < 0 \right\} = Q\left(\sqrt{2d\mathrm{SNR}}\right),
$$

with $\mathrm{SNR} = 1/(2\sigma^2)$, since y_i are Gaussians $\mathcal{N}(+1, \sigma^2)$, then $\sum_{i=1}^{d} 4y_i \sim \mathcal{N}(4d, 16d\sigma^2)$, $\Pr(X > x) = Q(\frac{x-\mu}{\sigma})$ and $Q(-x) = 1 - Q(x)$

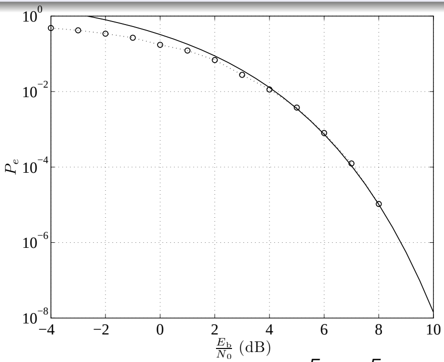
Error Probability and Union Bound

- Summarising we have that

$$P_e \leq \sum_d A_d Q\left(\sqrt{2d\,\mathrm{SNR}}\right) \qquad P_b \leq \sum_d \sum_i \frac{i}{k} A_{i,d} Q\left(\sqrt{2d\,\mathrm{SNR}}\right)$$

- Since Q is a decreasing function, at large SNR we have that

$$P_e \leq \sum_d A_d Q\left(\sqrt{2d\,\mathrm{SNR}}\right) \approx A_{d_{\min}} Q\left(\sqrt{2d_{\min}\,\mathrm{SNR}}\right).$$

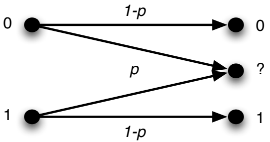


(7,4) Hamming code performance P_e vs $\frac{E_b}{N_0}$, $R\frac{E_b}{N_0} = \mathrm{SNR} = \frac{E_s}{N_0}$

Random Coding for the BEC

Coding and Decoding for the Binary Erasure Channel

- For the BEC, linear codes can be decoded by matrix inversion:
  - eliminate the columns of $\boldsymbol{G}$ corresponding to erased positions in the codeword $\longrightarrow \boldsymbol{G}'$
  - invert $\boldsymbol{G}'$
  - recover the information bits $\boldsymbol{b} = \boldsymbol{c}'\boldsymbol{G}'^{-1}$ where $\boldsymbol{c}'$ is the vector containing only the non-erased bits of the received sequence
- A similar decoder can be constructed based on the parity-check matrix $\boldsymbol{H}$, where decoding is achieved via triangulation of the portion of $\boldsymbol{H}$ corresponding to the erased bits
- The complexity of matrix inversion or triangulation decoding is the complexity of Gauss elimination over GF(2), i.e. on the order n^2 if n is the codeword length
- What is the probability of success of matrix inversion decoding if the generator matrix $\boldsymbol{G}$ has been selected at random? (random coding)



Binary Erasure Channel (BEC)

Random Coding for the BEC

Probability of Inverting a Random Matrix

- The matrix inversion decoder will be successful if the matrix $\boldsymbol{G}'$ with erased columns has rank $k = nR$, i.e., if $\boldsymbol{G}'$ has full rank
- Let $\boldsymbol{A}$ be a random binary $k \times n$ matrix chosen uniformly at random, with $k \leq n$. How probable is it that $\boldsymbol{A}$ has rank k?
- There are $2^{k \times n}$ binary $k \times n$ matrices and $\prod_{i=0}^{k-1}(2^n - 2^i)$ of them have rank k (for each row, choose any sequence of length n except any linear (binary) combination of previous rows)
- The resulting probability of full rank is

$$P(\mathrm{rank}(\boldsymbol{A}) = k) = \frac{\prod_{i=0}^{k-1}(2^n - 2^i)}{2^{k \times n}} = \prod_{i=n-k+1}^{n} (1 - 2^{-i})$$

- For $n = k$, we have

$$P(\mathrm{rank}(\boldsymbol{A}) = k) = \frac{1}{2}\frac{3}{4}\frac{7}{8}\frac{15}{16}\ldots(1 - 2^{-n})$$

  whose limit as n goes to infinity is 0.288788
- For $n > k$, the product omits the first and smallest terms $(1/2, 3/4$, etc.), so the limit gets larger and closer to 1 as $n - k$ grows

Random Coding for the BEC

Rate and Chebyshev's inequality

- Remember that the capacity of a BEC with erasure probability p is $C = 1 - p$ and we know from the converse to the coding theorem that we cannot hope to achieve arbitrary reliability for $R \geq C$ with any type of coding, so all the more so now that we restrict ourselves to linear coding
- Therefore, let the rate be $R = 1 - p - \varepsilon$ for any arbitrarily small $\varepsilon > 0$
- Let W be the number of erased bits in our block of length n. W follows a binomial distribution

$$P_W(w) = \binom{n}{w} p^w (1-p)^{n-w},$$

  and we have $\mathrm{E}[W] = np$ and $\mathrm{var}(W) = np(1-p)$
- We use Chebyshev's inequality

$$P(|W - pn| \geq \alpha) \leq \frac{np(1-p)}{\alpha^2},$$

  which, by setting $\alpha = \delta n$, gives us

$$P(|W - pn| \leq \delta n) \geq 1 - \frac{p(1-p)}{n\delta^2}.$$

Random Coding for the BEC

Probability of success for random coding

- Let us denote $D = |W - pn|$. We can now write the probability of successful decoding P_s as

$$\begin{aligned} P_s &= P_{s|D\leq\delta n}P(D \leq \delta n) + P_{s|D>\delta n}P(D > \delta n) \\ &\geq P_{s|D\leq\delta n}P(D \leq \delta n) && \text{(dropping a positive term)} \\ &\geq P_{s|W=pn+\delta n}\left(1 - \frac{p(1-p)}{n\delta^2}\right) && \text{(Chebyshev's inequality)} \end{aligned}$$

  where we have also used the fact that the probability of success over the interval $|W - pn| \leq \delta n$ is smallest[a] for $W = pn + \delta n$

- We now use the expression we computed for the probability of successfully inverting a random matrix, whose dimensions are $nR = n(1 - p - \varepsilon)$ rows and $n - (pn + \delta n) = n(1 - p - \delta)$ columns, to get

$$P_s \geq \left(1 - \frac{p(1-p)}{n\delta^2}\right) \prod_{i=n(\varepsilon-\delta)+1}^{n(1-p-\delta)} (1 - 2^{-i})$$

[a] we brush over all integer constraints on the number of erasures and the matrix sizes. The proof can be made precise by appropriate use of floor or ceiling integer rounding functions.

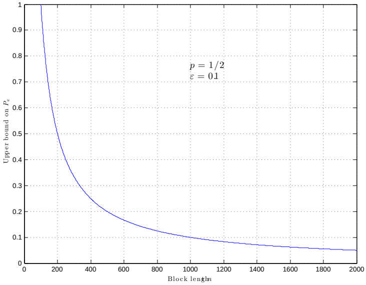
Random Coding for the BEC

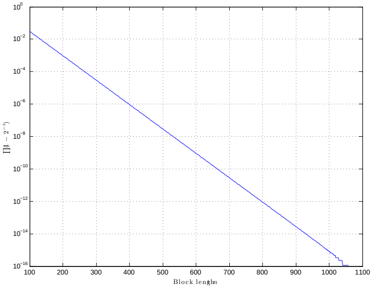
Probability of error for random coding

- We now get for the probability of error $P_e = 1 - P_s$, by choosing $\delta = \varepsilon/2$,

$$P_e \leq 1 - \left(1 - \frac{4p(1-p)}{n\varepsilon^2}\right) \prod_{i=n\varepsilon/2+1}^{n(1-p-\varepsilon/2)} (1 - 2^{-i})$$

  which can be made arbitrarily small for any given ε by choosing n appropriately large





Upper bounds including the Chebyshev averaging - excluding averaging (i.e. assuming $W = np$)

Random coding for the BEC

What we have learnt...

- For the BEC, linear codes achieve arbitrary reliability on average over all codes by choosing n large
- While the bound for a specific number of erasures is exponential in the block length, the overall bound we calculated is not: this comes from the Chebyshev averaging which is a weak bounding technique and can be improved by use of Chernoff or Gallager bounding
- In fact, linear codes achieve arbitrary reliability on average for all input-symmetric channels (we will not prove that) including the AWGN channel with BPSK that we studied earlier
- Linear coding provides a low-complexity method to define a set of codewords (better than picking 2^{nR} codewords at random) and to encode information digits via matrix multiplication
- What we need now is techniques for efficient decoding that work better than exhaustive search for the maximum likelihood solution