4F5: Advanced Communications and Coding

Coding Handout 4: Reed Solomon Codes

Jossy Sayir

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

Michaelmas 2014

t error correcting coding

Problem Statement

Design a linear code over a field $\mathcal{F}$ that can correct any pattern of up to and including t errors in a codeword of length N.

Source, code and observation alphabet (channel/demodulator output) assumed to be identical and of cardinality $\#(\mathcal{F})$. An "error" is defined as any occurrence where a received symbol is not equal to the corresponding transmitted symbol.

Compressed Sensing

This signal processing problem is related to the t error correcting coding problem on the previous slide.

Problem Statement

Given a signal vector $\boldsymbol{x}$ that is known to have only up to t non-zero components, design a minimal set of "sensors" in the form of linear combinations of elements of $\boldsymbol{x}$ from which $\boldsymbol{x}$ can be recovered in full.

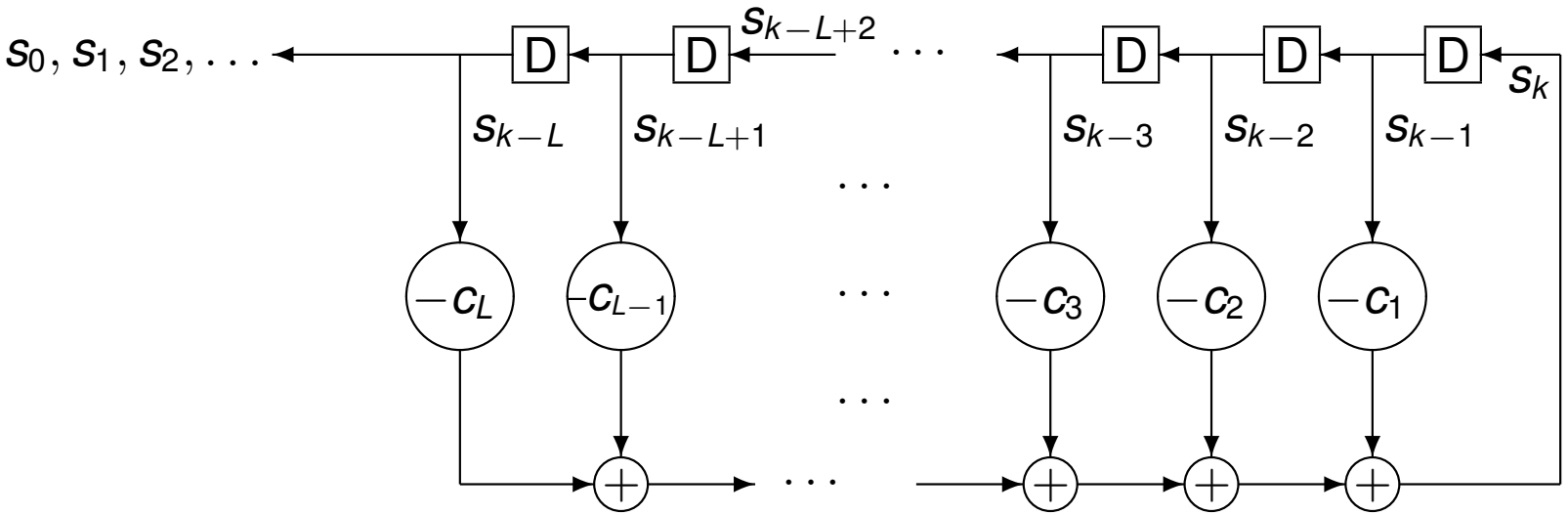
Hamming Weight (reminder)

Definition

The Hamming Weight of a vector $\boldsymbol{x}$, denoted $w(\boldsymbol{x})$, is the number of non-zero components of $\boldsymbol{x}$.

- Rephrasing the compressed sensing problem: how many linear combinations of all components of $\boldsymbol{x}$ do I need in order to recover $\boldsymbol{x}$ in full given that $w(\boldsymbol{x}) \leq t$?
- Rephrasing the t error correction problem: let $\boldsymbol{x}$ be the transmitted codeword and $\boldsymbol{r} = \boldsymbol{x} + \boldsymbol{e}$ be the received word. Design a code such that all errors will be corrected if $w(\boldsymbol{e}) \leq t$.

Linear Feedback Shift Register (LFSR)



- $s_0, s_1, \ldots, s_{L-1}$ can be chosen at will and constitute the initial state of the LFSR
- for $k = L, L+1, L+2, \ldots$, the following relation holds

$$s_k + \sum_{i=1}^{L} c_i s_{k-i} = 0.$$

LFSR and the z-transform (also called D-transform)

- for a sequence $\boldsymbol{s} = s_0, s_1, s_2, \ldots$ defined over any field $\mathcal{F}$, the z transform is

$$S(z) = \sum_{i=0}^{\infty} s_i z^{-i}$$

- in the coding literature, the D-transform is often used, where $D = z^{-1}$. The advantage of using D is that one can talk about "degree of a polynomial" in the usual way. We will stick to the z-transform in the interest of consistency with 3F1, but when we speak of "degree" we will mean the degree of a polynomial in the variable z^{-1}, or the negative power of largest magnitude.
- the LFSR relation

$$s_k + \sum_{i=1}^{L} c_i s_{k-i} = 0 \text{ for } k \geq L$$

transforms in the z-domain as

$$C(z)S(z) = P(z)$$

where $C(z) = 1 + c_1 z^{-1} + \ldots + c_L z^{-L}$, $S(z)$ is the z-transform of the LFSR output sequence, and $P(z)$ is a polynomial in z^{-1} of degree at most $L-1$, i.e., zero coefficients for entries corresponding to times $L, L+1, L+2, \ldots$

Linear Complexity

Definition

The linear complexity $\mathcal{L}(\boldsymbol{s})$ of any sequence $\boldsymbol{s} = s_0, s_1, s_2, \ldots$ is the length of the shortest LFSR that can reproduce the sequence.

- z-transform interpretation: let $S(z) = \sum_{i=0}^{\infty} s_i z^{-i}$, then if $C(z)$ is the polynomial of smallest degree such that

$$S(z) = \frac{P(z)}{C(z)}, \text{ with } \deg(P(z)) < \deg(C(z)),$$

  then $\mathcal{L}(\boldsymbol{s}) = \deg(C(z))$.
- for finite length sequences, the linear complexity is defined as that of the shortest LFSR that can reproduce the finite length sequence irrespective of what follows

Discrete Fourier Transform (DFT)

Definition

Let α be a primitive N-th root of unity in $\mathcal{F}$, i.e., $\alpha^0, \alpha^1, \alpha^2, \ldots, \alpha^{N-1}$ are all distinct and $\alpha^N = \alpha^0 = 1$, then the discrete Fourier transform $\boldsymbol{X} = (X_0, \ldots, X_{N-1})$ of a vector $\boldsymbol{x} = (x_0, \ldots, x_{N-1}) \in \mathcal{F}^N$ is defined by

$$X_k = \sum_{i=0}^{N-1} x_i \alpha^{ik}$$

and the inverse transform is defined by

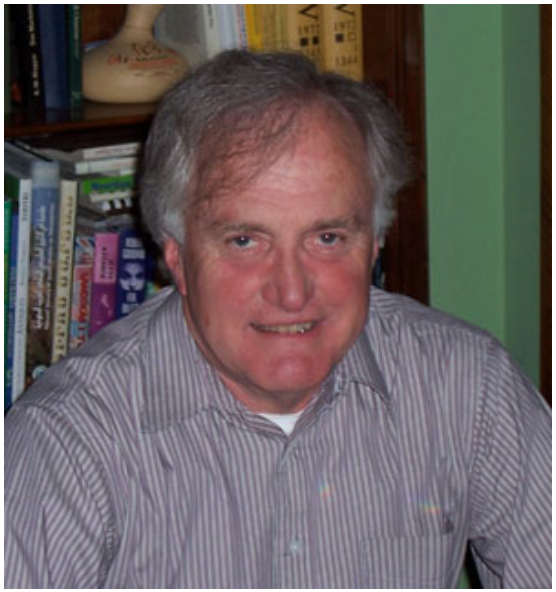
$$x_i = \frac{1}{N^\star} \sum_{k=0}^{N-1} X_k \alpha^{-ik}$$

where $N^\star = N$ for $\mathcal{F} = \mathbb{C}$ or $\mathbb{R}$ or for prime finite fields GF(p), and $N^\star = N$ modulo p for extension finite fields GF(p^m).

Discrete Fourier Transform (DFT)

- if $\mathcal{F} = \mathbb{C}$ then we obtain the familiar discrete Fourier transform with $\alpha = e^{-2j\pi/N}$, which can be defined for any desired length $N > 1$.
- if $\mathcal{F} = \mathbb{R}$ there exists only one discrete Fourier transform of length 2 for $\alpha = -1$.
- for finite fields, discrete Fourier transforms can only exist for N divides the order $p - 1$ of the multiplicative group (or $p^m - 1$ for extension fields) due to Lagrange's theorem. Any group element except 1 can be used as the α to define a discrete Fourier transform whose length will be the order of that element.

"The Theorem that Massey attributed to Blahut"



Jim Massey

This theorem was never stated as such by Blahut but was coined "Blahut's theorem" by Jim Massey in a book review and in his lecture notes.

Dick Blahut

Theorem

Let $\boldsymbol{s}$ be a vector over $\mathcal{F}$ and $\boldsymbol{S}$ be its N point discrete Fourier transform, then, provided $w(\boldsymbol{s}) < N/2$,

$$w(\boldsymbol{s}) = \mathcal{L}(\boldsymbol{S}),$$

i.e., the Hamming weight of a time-domain vector is equal to the linear complexity of its discrete Fourier transform.

Proof

Let $\bar{\boldsymbol{S}}$ be the periodic repetition of $\boldsymbol{S}$, then

$$\begin{aligned}\sum_{i=0}^{\infty} \bar{S}_i z^{-i} &= \sum_{i=0}^{\infty} S_{i\%N} z^{-i} \\ &= \sum_{i=0}^{\infty} \sum_{k=0}^{N-1} s_k \alpha^{k(i\%N)} z^{-i} = \sum_{i=0}^{\infty} \sum_{k=0}^{N-1} s_k \alpha^{ki} z^{-i} \\ &= \sum_{k=0}^{N-1} s_k \sum_{i=0}^{\infty} (\alpha^k z^{-1})^i \\ &= \sum_{k=0}^{N-1} \frac{s_k}{1-\alpha^k z^{-1}} \end{aligned} \tag{1}$$

where we used the notation $i\%N$ for i modulo N. The final expression is a proper partial fraction expansion of a rational function with $w(\boldsymbol{s})$ distinct roots, and hence can be re-written as quotient of polyomials $P(z)/C(z)$ where $P(z)$ has degree at most $w(\boldsymbol{s}) - 1$ and $C(z)$ has degree $w(\boldsymbol{s})$.

LFSR Synthesis

- Given a sequence $\boldsymbol{s}$, how do we synthesise a minimal LFSR that generates $\boldsymbol{s}$?
- Problem can be seen as solving linear equations in unknowns $c_1, c_2, \ldots, c_L$ where $L = \mathcal{L}(\boldsymbol{s})$, i.e.,

$$\begin{cases} s_{L+1} & +c_1 s_L + \ldots + c_L s_1 = 0 \\ s_{L+2} & +c_1 s_{L+1} + \ldots + c_L s_2 = 0 \\ \vdots & \vdots \\ s_{2L} & +c_1 s_{2L-1} + \ldots + c_L s_L = 0. \end{cases}$$

  If the observation window for $\boldsymbol{s}$ has length at least $2\mathcal{L}(\boldsymbol{s})$, then there are at least $\mathcal{L}(\boldsymbol{s})$ equations for the $\mathcal{L}(\boldsymbol{s})$ unknowns and the desired LFSR can be syntesised through matrix inversion
- Berlekamp-Massey algorithm synthesises an LFSR of length $\mathcal{L}(\boldsymbol{s})$ efficiently after observing at least $2\mathcal{L}(\boldsymbol{s})$ elements of $\boldsymbol{s}$
- $2\mathcal{L}(\boldsymbol{s})$ observations of $\boldsymbol{s}$ are sufficient to generate the whole sequence $\boldsymbol{s}$

Compressed Sensing

Question (reminder)

How many measurements (linear combinations) of $\boldsymbol{x}$ do I need in order to recover $\boldsymbol{x}$ in full given that $w(\boldsymbol{x}) \leq t$?

Answer

Take $2t$ linear combinations corresponding to any $2t$ consecutive rows of the DFT matrix, thereby obtaining $2t$ Fourier coefficients of $\boldsymbol{x}$. Synthesise the LFSR of length $\mathcal{L}(\boldsymbol{X}) = t$ that recovers $\boldsymbol{X}$. Take the inverse DFT to recover $\boldsymbol{x}$.

Warning

- in theory, this works for any field including $\mathbb{C}$
- in practice, an LFSR is not a stable system over continuous fields and hence this method will *not* work over $\mathbb{C}$
- "Compressed Sensing" is a major topic of ongoing research for the signal processing community. Best recovery results are obtained using L1 minimisation and typically require at least $5t$ measurements (linear combinations)

Reed Solomon (RS) Codes

- Parity-Check matrix, for any even $M < N$,

$$\boldsymbol{H} = \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \dots & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \dots & \alpha^{N-1} \\ \alpha^0 & \alpha^2 & \alpha^4 & \dots & \alpha^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha^0 & \alpha^M & \alpha^{2M} & \dots & \alpha^{M(N-1)} \end{bmatrix} = [\alpha^{ij}] \text{ for } i = 0, \dots, M-1 \text{ and } j = 0, \dots, N-$$

- $\boldsymbol{H}$ consists of the first M rows of the DFT matrix
- $\boldsymbol{x}\boldsymbol{H}^T = 0$ implies that the first M coefficients the discrete Fourier transform of any codeword are zero, i.e., a codeword is any sequence over $\mathcal{F}$ whose spectrum vanishes over a band of frequencies
  - It can be shown that $\boldsymbol{H}$ has full rank and hence an RS code has rate $R = (N - M)/N$ with $|\mathcal{F}|^{N-M}$ codewords of length N
  - For example, one possible RS code for ISO 18004:2006 (QR codes) over GF(2^8) with $N = 255$ and $M = 12$ has length 255 bytes or, equivalently, 2040 bits, and encodes 243 bytes or, equivalently, 1944 bits, for a rate of $R = 0.953$

Decoding RS Codes

- The received word $\boldsymbol{r} = \boldsymbol{x} + \boldsymbol{e}$ is the codeword plus the error sequence Since the DFT is linear, this relation holds in the frequency domain, i.e., $\boldsymbol{R} = \boldsymbol{X} + \boldsymbol{E}$. We know that the DFT of the codeword has M zero coefficients. Hence, the corresponding M coefficients of $\boldsymbol{R}$ are the coefficients of the error vector $\boldsymbol{E}$
- Assuming that the error sequence contains at most t errors, the linear complexity of $\boldsymbol{E}$ is at most t and hence observing $2t = M$ entries of $\boldsymbol{E}$ suffices to reconstruct $\boldsymbol{E}$ and, by inverse DFT, reconstruct $\boldsymbol{e}$
- Algorithm:
  1. compute $\boldsymbol{r}\boldsymbol{H}^T = (\boldsymbol{x} + \boldsymbol{e})\boldsymbol{H}^T = \boldsymbol{e}\boldsymbol{H}^T = (E_0, \ldots, E_{M-1})$
  2. synthesise the LFSR that generates $\boldsymbol{E} = (E_0, \ldots, E_N)$ using matrix inversion or the Berlekamp-Massey algorithm
  3. take the inverse DFT of $\boldsymbol{E}$ to obtain $\boldsymbol{e}$
  4. compute $\boldsymbol{x} = \boldsymbol{r} - \boldsymbol{e}$
- if the error sequence has t or less errors, the decoder is guaranteed to correct them irrespective of their position in the received word
- for example, the (255,243) RS code over GF(2^8) can correct any pattern of 6 errors as the DFT of its codewords vanishes over 12 coefficients

RS Code example

- RS code over GF(11). Multiplicative group has order 10, so DFT can have length 2, 5 or 10. We want a code of length 10, so pick an element of order 10.
- $2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 5, 2^5 = 10, 2^6 = 9, 2^7 = 7, 2^8 = 3, 2^9 = 6$, so we can use $\alpha = 2$
- Say we want to correct up to 2 errors, so $M = 4$, and

$$\boldsymbol{H} = \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 \\ \alpha^0 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} & \alpha^{12} & \alpha^{14} & \alpha^{16} & \alpha^{18} \\ \alpha^0 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} & \alpha^{18} & \alpha^{21} & \alpha^{24} & \alpha^{27} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 5 & 10 & 9 & 7 & 3 & 6 \\ 1 & 4 & 5 & 9 & 3 & 1 & 4 & 5 & 9 & 3 \\ 1 & 8 & 9 & 6 & 4 & 10 & 3 & 2 & 5 & 7 \end{bmatrix}$$

RS Code example (continued)

- Systematic parity-check and encoder matrix

$$\boldsymbol{H}_{\text{sys}} = \begin{bmatrix} 2 & 8 & 2 & 7 & 5 & 6 & 1 & 0 & 0 & 0 \\ 10 & 9 & 7 & 4 & 10 & 2 & 0 & 1 & 0 & 0 \\ 7 & 5 & 5 & 4 & 5 & 9 & 0 & 0 & 1 & 0 \\ 4 & 1 & 9 & 8 & 3 & 6 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{G}_{\text{sys}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 9 & 1 & 4 & 7 \\ 0 & 1 & 0 & 0 & 0 & 0 & 3 & 2 & 6 & 10 \\ 0 & 0 & 1 & 0 & 0 & 0 & 9 & 4 & 6 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 & 4 & 7 & 7 & 3 \\ 0 & 0 & 0 & 0 & 1 & 0 & 6 & 1 & 6 & 8 \\ 0 & 0 & 0 & 0 & 0 & 1 & 5 & 9 & 2 & 5 \end{bmatrix}$$

- Say we want to encode $\boldsymbol{u} = (7, 8, 8, 4, 7, 1)$, we have

$$\boldsymbol{x} = \boldsymbol{u}\boldsymbol{G}_{\text{sys}} = (7, 8, 8, 4, 7, 1, 2, 0, 9, 9)$$

and we can verify that $\boldsymbol{x}\boldsymbol{H}^T = \boldsymbol{x}\boldsymbol{H}_{\text{sys}}^T = 0$ as expected

RS Code example (continued)

- Now suppose that $\boldsymbol{r} = (7, 8, 3, 4, 7, 1, 2, 0, 4, 9)$ was received
- Compute the syndrome, corresponding to the first 4 Fourier coefficients of $\boldsymbol{r}$,

$$\boldsymbol{s} = \boldsymbol{r}\boldsymbol{H}^T = (1, 9, 7, 7)$$

- find the coefficients (c_0, c_1) of a LFSR of length 2 that generates $\boldsymbol{s}$ by solving the equations

$$\begin{cases} 7 + 9c_0 + 1c_1 &= 0 \\ 7 + 7c_0 + 9c_1 &= 0 \end{cases}$$

  yielding $(c_0, c_1) = (4, 1)$
- reconstruct the complete DFT of the error sequence using the LFSR, $\boldsymbol{E} = (1, 9, 7, 7, 9, 1, 9, 7, 7, 9)$
- take the inverse DFT to obtain $\boldsymbol{e} = (0, 0, 6, 0, 0, 0, 0, 0, 6, 0)$
- recover the transmitted codeword

$$\begin{aligned} \boldsymbol{x} = \boldsymbol{r} - \boldsymbol{e} &= (7, 8, 3, 4, 7, 1, 2, 0, 4, 9) - (0, 0, 6, 0, 0, 0, 0, 0, 6, 0) \\ &= (7, 8, 8, 4, 7, 1, 2, 0, 9, 9) \end{aligned}$$

- cut out systematic part to recover the information symbols $\boldsymbol{u} = (7, 8, 8, 4, 7, 1)$

Distance Properties of RS Codes

- A Reed-Solomon code is an (N, K) linear code that can correct any pattern of up to t errors and hence its minimum distance, which must be strictly larger than $2t$, satisfies
$$d_{\min} \geq 2t + 1 = M + 1 = N - K + 1.$$
- By the Singleton bound, the minimum distance of any (N, K) code must satisfy
$$d_{\min} \leq N - K + 1$$

Minimum Distance

The mininum distance of a Reed-Solomon code is $d_{\min} = N - K + 1$. The code is maximum distance separable.

Interpolation and Decimation

$\boldsymbol{x}$ a block of 100 samples of signal $x(t)$ of bandwidth 5 Hz, sampled at 20 samples per second

Interpolation

- *Problem:* Can we find the 200 samples we would have had if we sampled $x(t)$ at 40 samples per second?
- *Solution:* Take DFT, insert 100 zeros, take inverse DFT.
- Sampling theorem tells us that resulting vector of length 200 is exactly the vector we would have had if $x(t)$ had been sampled at 40 samples per second.

Decimation

- *Problem:* Can we find the 50 samples we would have had if we sampled $x(t)$ at 10 samples per second?
- *Solution:* Drop every second sample. Alternatively, take DFT, eliminate 50 coefficients around the middle, then take inverse DFT.
- Sampling theorem states that this works if the bandwidth of $x(t)$ is less than half of the new sampling rate.

RS Codes

There is an interesting parallel between Reed Solomon codes and interpolation in signal processing. Reed Solomon codes can be seen as the finite field equivalent of re-sampling the information sequence at a higher sampling frequency.