IERG6120 Coding for Distributed Storage Systems Lecture 12 - 05/11/2016

Berlekamp-Massey decoding of RS code

Lecturer: Kenneth Shum *Scribe: Bowen Zhang*

1 Berlekamp-Massey algorithm

We recall some notations from lecture 11. Let $(S_1, S_2, S_3, \ldots, S_n)$ be an arbitrary sequence of elements in a field K. We say that this sequence is described by a linear feedback shift register (LFSR) of length L if there exists L field elements $\Lambda_1, \Lambda_2, \ldots, \Lambda_L$, such that S_i can be obtained as a linear function of the previous L elements,

$$S_i = -\Lambda_1 S_{i-1} - \Lambda_2 S_{i-2} - \cdots - \Lambda_L S_{i-L} \tag{1}$$

for $i = L+1, L+2, \ldots, n$. We specify the linear feedback shift register by a pair $(L, \Lambda(Z))$, where L denotes the length of the LFSR and $\Lambda(Z)$ is the *feedback polynomial* defined as

$$\Lambda(Z) := 1 + \Lambda_1 Z + \Lambda_2 Z^2 + \cdots + \Lambda_L Z^L.$$

We note that the degree of $\Lambda(Z)$ is less than or equal to L, and the constant term is equal to 1. (The degree is strictly less than L if $\Lambda_L = 0$.)

For notational convenience, we define $\Lambda_0 := 1$, so that (1) can be written compactly as

$$0 = \sum_{j=0}^{L} \Lambda_j S_{i-j},$$

for $i = L+1, \ldots, n$.

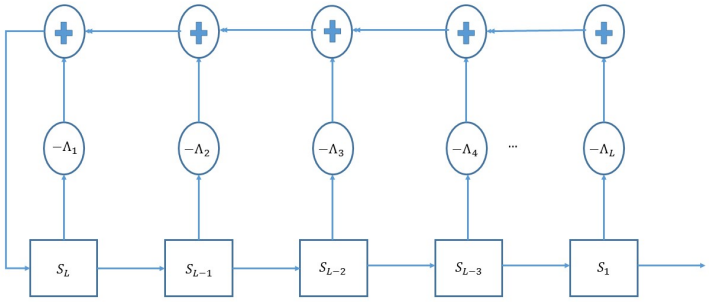

Figure 1: Example of linear feedback shift register

We want to find the shortest LFSR that generates $(S_1, S_2, S_3, \ldots, S_n)$. The length of the *shortest* LFSR that generates $(S_1, S_2, S_3, \ldots, S_n)$ is called the *linear complexity* of $(S_1, S_2, S_3, \ldots, S_n)$.

For $M = 1, 2, \ldots, n$, let L_M be the linear complexity of the first M terms $(S_1, S_2, \ldots, S_M)$, and let $\Lambda^M(Z) = \sum_{j=0}^{L_M} \Lambda_j^{M-1} Z^j$ be the corresponding feedback polynomial.

Caveat: There may be more than one LFSR that achieve the shortest length.

We have the following chain of inequalities

$$L_1 \leq L_2 \leq L_3 \leq \cdots \leq L_n, \tag{2}$$

because, if $(L_M, \Lambda^M(Z))$ is the shortest LFSR that generates the first M terms, it also generates the first $M-1$ terms. The non-decreasing sequence of integers in (2) is usually called the *linear complexity profile* of $(S_1, S_2, S_3, \ldots, S_n)$.

The M-th term produced by $(L_{M-1}, \Lambda^{M-1}(Z))$,

$$\hat{S}_M := -\sum_{j=1}^{L_{M-1}} \Lambda_j^{M-1} S_{M-j}$$

may not equal to the desired value S_M. We let

$$\Delta_M := S_M - \hat{S}_M = \sum_{j=0}^{L_{M-1}} \Lambda_j^{M-1} S_{M-j},$$

be the difference between S_M and $\hat{S}_M$. If $\Delta_M = 0$, then the LFSR $(L_{M-1}, \Lambda^{M-1}(Z))$ correctly computes the M-th term S_M, and we can let $(L_M, \Lambda^M(Z)) = (L_{M-1}, \Lambda^{M-1}(Z))$. When $\Delta_M \neq 0$, we have shown in lecture 11 that

Theorem 1. *Suppose that $(L_i, \Lambda^i(Z))$ is the shortest LFSR that produces $(S_1, S_2, \ldots, S_i)$, for $i = 1, 2, \ldots, M$. If $\Delta_M \neq 0$, then*

$$L_M \geq \max(L_{M-1}, M - L_{M-1}).$$

The Berlekamp-Massey algorithm computes the linear complexity profile and the corresponding feedback polynomials of a sequence of elements $(S_1, \ldots, S_n)$ from a field K.

The algorithm computes $L_1, L_2, \ldots$ iteratively. In each step, we consider two cases.

If $\Delta_M = 0$, set $L_M = L_{M-1}$ and $\Lambda^M(Z) = \Lambda^{M-1}(Z)$.

If $\Delta_M \neq 0$, then the feedback polynomial $\Lambda^M(Z)$ is obtained by

$$\Lambda^M(Z) = \Lambda^{M-1}(Z) + \Lambda^{\mu-1}(Z) Z^e \alpha,$$

where $\mu \leq M$, $e \in \mathbb{Z}$, and $\alpha \in K$. The value of μ, e and α are obtained by the following theorem.

Theorem 2. *Suppose $(L_i, \Lambda^i(Z))$ is the shortest LFSR for $(S_1, \ldots, S_i)$, satisfying*

$$L_i = \begin{cases} L_{i-1} & \text{if } \Delta_i = 0, \\ \max(L_{i-1}, i - L_{i-1}) & \text{if } \Delta_i \neq 0, \end{cases}$$

for $i = 1, 2, 3, \ldots, M-1$. If $\exists \mu < M$, satisfying

$$L_{\mu-1} < L_\mu = L_{\mu+1} = \ldots = L_{M-1},$$

and $\Delta_\mu \neq 0$, then we can find an LFSR $(L_M, \Lambda^M(Z))$ that generates $(S_1, \ldots, S_M)$, with length

$$L_M = \begin{cases} L_{M-1} & \text{if } \Delta_M = 0, \\ \max(L_{M-1}, M - L_{M-1}) & \text{if } \Delta_M \neq 0. \end{cases}$$

Proof If $\Delta_M = 0$, we set $\Lambda^M(Z) = \Lambda^{M-1}(Z)$ and $L_M = L_{M-1}$.

Suppose that $\Delta_M \neq 0$. Set

$$\Lambda^M(Z) = \Lambda^{M-1}(Z) - \frac{\Delta_M}{\Delta_\mu} Z^{M-\mu} \Lambda^{\mu-1}(Z). \tag{3}$$

We remark that we do not have division by zero in (3), because Δ_μ is non-zero by the assumption in the theorem. Let

$$D = \deg \Lambda^M(Z) \leq \max(L_{M-1}, M - \mu + L_{\mu-1})$$

be the degree of $\Lambda^M(Z)$ defined in (3).

We check that the LFSR specified by feedback polynomial $\Lambda^M(Z)$ can generate $(S_1, \ldots, S_M)$. By the assumptions in the theorem, we have

$$\sum_{i=0}^{L_{\mu-1}} \Lambda_i^{\mu-1} S_{j-i} = \begin{cases} 0 & \text{if } j = L_{\mu-1}+1, L_{\mu-1}+2, \ldots, \mu-1, \\ \Delta_\mu & \text{if } j = \mu, \end{cases}$$

and

$$\sum_{i=0}^{L_{M-1}} \Lambda_i^{M-1} S_{j-i} = \begin{cases} 0 & \text{if } j = L_{M-1}+1, L_{M-1}+2, \ldots, M-1, \\ \Delta_M & \text{if } j = M. \end{cases}$$

With the notations $\Lambda_i^{M-1} = 0$ for $i = L_{M-1}+1, L_{M-1}+2, \ldots, D$ and $\Lambda_i^{\mu-1} = 0$ for $i = L_{\mu-1}+1, L_{\mu-1}+2, \ldots, D$, we can write the recursion as

$$\sum_{i=0}^{D} \Lambda_i^M S_{j-i} = \sum_{i=0}^{D} \Lambda_i^{M-1} S_{j-i} - \frac{\Delta_M}{\Delta_\mu} \sum_{i=0}^{D} \Lambda_i^{\mu-1} S_{j-M+\mu-i}. \tag{4}$$

If $j = M$, then the two summations on the right-hand side of (4) are equal to $\sum_{i=0}^{D} \Lambda_i^{M-1} S_{M-i} = \Delta_M$ and

$$\sum_{i=0}^{D} \Lambda_i^{\mu-1} S_{M-M+\mu-i} = \sum_{i=0}^{D} \Lambda_i^{\mu-1} S_{\mu-i} = \Delta_\mu,$$

respectively. Therefore, the right-hand side of (4) is equal to zero.

Now suppose that $j < M$. The first summation on the right-hand side of (4) is equal to zero for $j > L_{M-1}$, and the second summation is equal to zero if

$$j - M + \mu > L_{\mu-1},$$

By the hypothesis that $\Delta_\mu \neq 0$, we have $L_\mu = \mu - L_{\mu-1}$. Hence the second summation in (4) is equal to zero when

$$j > M - \mu + L_{\mu-1} = M - L_\mu = M - L_{\mu+1} = \cdots = M - L_{M-1}.$$

We conclude that $\sum_{i=0}^{D} \Lambda_i^M S_{j-i}$ is equal to zero for

$$\max(L_{M-1}, M - L_{M-1}) < j \leq M.$$

We can set $L_M = \max(L_{M-1}, M - L_{M-1})$. The degree of Λ^M is no more than

$$\max(L_{M-1}, M - \mu + L_{\mu-1}) = \max(L_{M-1}, M - L_{M-1}) = L_M.$$

The polynomial $\Lambda^M(Z)$ specifies an LFSR with length no more than L_M. The LFSR $(L_M, \Lambda^M(Z))$ generates $(S_1, S_2, \ldots, S_M)$. □

Given a sequence $(S_1, S_2, \ldots, S_N)$, we find the smallest index m such that $S_m \neq 0$. We initialize the algorithm by

$$L_j = 0,\ \Lambda^j(Z) = 1,$$

for $j = 0, 1, \ldots, m-1$, and

$$L_m = m,\ \Lambda^m(Z) = 1.$$

The LFSR $(L_i, \Lambda^i(Z))$ satisfy the conditions in Theorem 1 with

$$\Delta_{m-1} = S_m \neq 0.$$

The algorithm continues with repeated applications of Theorem 1 for $M = m, m+1, m+2, \ldots, n$.

Example. Determine the linear complexity profile of the sequence

$$S_1 = 0,\ S_2 = 1,\ S_3 = 0,\ \text{ and } S_i = 1 \text{ for } i \geq 4,$$

over $\mathbb{F}_2$. This sequence is ultimately periodic.

Initialization. The first non-zero element occurs at $S_2 = 1$. Let $(L_0, \Lambda^0(Z)) = (L_1, \Lambda^1(Z)) = (0, 1)$, and $(L_2, \Lambda^2(Z)) = (2, 1)$. We have $\Delta_2 = 1$.

$M = 3$. As $\Delta_3 = 0$, we set $L_3 = L_2$ and $\Lambda^3(Z) = \Lambda^2(Z)$.

$M = 4$. $\Delta_4 = 1$. Set $L_4 = \max(L_3, 4 - L_3) = \max(2, 4-2) = 2$, and

$$\Lambda^4(Z) = \Lambda^3(Z) + Z^2\Lambda^1(Z) = 1 + Z^2.$$

$M = 5$. $\Delta_5 = 1$. Set $L_5 = \max(L_4, 5 - L_4) = \max(2, 5-2) = 3$, and

$$\Lambda^5(Z) = \Lambda^4(Z) + Z^4\Lambda^1(Z) = 1 + Z^2 + Z^3.$$

$M = 6$. Because $\Delta_6 = 0$, $(L_6, \Lambda^6(Z)) = (L_5, \Lambda^5(Z))$.

$M = 7$. $\Delta_7 = 1$. Set $L_7 = \max(L_6, 7 - L_6) = \max(3, 7-3) = 4$, and

$$\Lambda^7(Z) = \Lambda^6(Z) + Z^2\Lambda^4(Z) = (1 + Z^2 + Z^3) + Z^2(1 + Z^2) = 1 + Z^3 + Z^4.$$

$M = 8$. $\Delta_8 = 1$. Set $L_8 = \max(L_7, 8 - L_7) = \max(4, 8-4) = 4$, and

$$\Lambda^8(Z) = \Lambda^7(Z) + Z\Lambda^6(Z) = (1 + Z^4 + Z^3) + Z(1 + Z^2 + Z^3) = 1 + Z.$$

$M \geq 9$. $\Delta_M = 0$, $(L_M, \Lambda^M(Z)) = (L_8, \Lambda^8(Z)) = (4, 1 + Z)$.

The calculations are summarized in the following table.

i	0	1	2	3	4	5	6	7	8	9
S_i		0	1	0	1	1	1	1	1	1
Δ_i		0	1	0	1	1	0	1	1	0
L_i	0	0	2	2	2	3	3	4	4	4
$\Lambda^i(z)$	1	1	1	1	$1 + Z^2$	$1 + Z^2 + Z^3$	$1 + Z^2 + Z^3$	$1 + Z^3 + Z^4$	$1 + Z$	$1 + Z$

The linear complexity profile is 0,2,2,2,3,3,4,4,4,....

2 Decoding RS Codes

First we define some notations. Fix n distinct elements $\alpha_1, \ldots, \alpha_n$ in F_q. Let

$$g(Z) := (Z-\alpha_1)(Z-\alpha_2)\cdots(Z-\alpha_n)$$
$$g_i(Z) := \frac{g(Z)}{Z-\alpha_i} = (Z-\alpha_1)(Z-\alpha_2)\cdots(Z-\alpha_{i-1})(Z-\alpha_{i+1})(Z-\alpha_{i+2})\cdots(Z-\alpha_n).$$

We first prove the following useful lemma.

Lemma 3. *We have*

$$\sum_{i=1}^{n} \frac{\alpha_i^j}{g_i(\alpha_i)} = \begin{cases} 0 & \text{if } j = 0, 1, \ldots, n-2 \\ 1 & \text{if } j = n-1. \end{cases}$$

Proof We use the notation

$$V(\alpha_1, \ldots, \alpha_n) := \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{n-1} & \alpha_2^{n-1} & \cdots & \alpha_n^{n-1} \end{bmatrix} = \prod_{j>i} (\alpha_j - \alpha_i)$$

for the determinant of a Vandermonde matrix.

Suppose that we replace the last row of the above Vandermonde matrix by $[\alpha_i^j]_{i=1,\ldots,n}$, for some j between 1 and $n-1$,

$$\delta_j := \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{n-2} & \alpha_2^{n-2} & \cdots & \alpha_n^{n-2} \\ \alpha_1^{j} & \alpha_2^{j} & \cdots & \alpha_n^{j} \end{vmatrix}$$

Expand the determinant on the last row. We get

$$\begin{aligned} \delta_j &= \sum_{i=1}^{n} (-1)^{n+i} \alpha_i^j \cdot V(\alpha_1, \alpha_2, \ldots, \widehat{\alpha_i}, \ldots, \alpha_n) \\ &= \sum_{i=1}^{n} (-1)^{n+i} \alpha_i^j \cdot \frac{V(\alpha_1, \alpha_2, \ldots, \alpha_n)}{(\alpha_n - \alpha_i)\ldots(\alpha_{i+1} - \alpha_i)\cdot(\alpha_i - \alpha_{i-1})\cdots(\alpha_i - \alpha_1)} \\ &= V(\alpha_1, \alpha_2, \ldots, \alpha_n) \sum_{i=1}^{n} \frac{\alpha_i^j}{g_i(\alpha_i)}. \end{aligned}$$

$\therefore$ If $j = n-1$, then the determinant δ_j is equal to $V(\alpha_1, \ldots, \alpha_n)$, and we get $\sum_{i=1}^{n} \alpha_i^{n-1}/g_i(\alpha_i) = 1$. For $j = 0, 1, 2, \ldots, n-2$, we have $\delta_j = 0$, because there are two repeated rows in the determinant δ_j. Hence $\sum_{i=1}^{n} \alpha_i^j/g_i(\alpha_i) = 0$ for $j = 0, 1, 2, \ldots, n-2$. □

We consider an (n, k) Reed-Solomon codes with the following $k \times n$ generator matrix

$$G = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \cdots & \alpha_n^{k-1} \end{bmatrix} \tag{5}$$

where α_i's are distinct nonzero elements in a finite field $\mathbb{F}_q$. In the following, we need the assumption that all α_i's are non-zero, so that α_i^{-1} exists in $\mathbb{F}_q$ for all i. A list of k message symbols, m_1 to m_k, are encoded to a codeword by multiplying

$$(m_1, \ldots, m_k) \cdot G.$$

From Lemma 3, we can write down a parity-check matrix as

$$H = \begin{bmatrix} \frac{1}{g_1(\alpha_1)} & \frac{1}{g_2(\alpha_2)} & \cdots & \frac{1}{g_n(\alpha_n)} \\ \frac{\alpha_1}{g_1(\alpha_1)} & \frac{\alpha_2}{g_2(\alpha_2)} & \cdots & \frac{\alpha_n}{g_n(\alpha_n)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\alpha_1^{n-k-1}}{g_1(\alpha_1)} & \frac{\alpha_2^{n-k-1}}{g_2(\alpha_2)} & \cdots & \frac{\alpha_n^{n-k-1}}{g_n(\alpha_n)} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{n-k-1} & \alpha_2^{n-k-1} & \cdots & \alpha_n^{n-k-1} \end{bmatrix} \cdot diag(\frac{1}{g_1(\alpha_1)}, \ldots, \frac{1}{g_n(\alpha_n)}). \tag{6}$$

We note that $g_i(\alpha_i)$ are nonzero for all i, because the elements $\alpha_1, \ldots, \alpha_n$ are distinct.

The following is a syndrome-based method for decoding RS code.

Step(1). Calculate syndromes by multiplying the received vector Y and the transpose of parity-check matrix in (6).

The syndromes are the components of $yH^T = (s_1, s_2, ..., s_{d-1})$, where $d = n - k + 1$ is the minimum distance.

Step(2). Obtain the shortest linear feedback shift register $(L, \Lambda(Z))$ that generates the syndrome sequence $s_1, s_2, \ldots, s_{d-1}$.

Step(3). If the number of errors t is less than or equal to $\left\lfloor \frac{(d-1)}{2} \right\rfloor$, then the feedback polynomial $\Lambda(Z)$ has degree t and t distinct roots. There is an error at location i if and only if $\Lambda(\alpha_i^{-1}) = 0$. In the case when $\Lambda(Z)$ has no root in $\mathbb{F}_q$ or the number distinct roots of $\Lambda(Z)$ is strictly less than the degree of $\Lambda(Z)$, then we can declare that there are more than $\left\lfloor \frac{(d-1)}{2} \right\rfloor$ errors, and stop the decoding procedure.

Step(4). After locating the errors, then error values can be calculated by solving a system of linear equations.

Exercise.

1. In the last step of the decoding procedure of RS code, we need to determine the error values. Suppose that we have already determined the location of the errors, and they are $i_1 < i_2 < \cdots < i_t$ for some integer t less than or equal to $\left\lfloor \frac{(d-1)}{2} \right\rfloor$.

Let e be the error vector, defined as the difference between the received vector and the transmitted codeword. Let the i_j-th component of e be e_{i_j}, for $j = 1, 2 \ldots, t$. Show that the error values e_{i_j} satisfy the following system of linear equations:

$$\begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_t \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha_{i_1} & \alpha_{i_2} & \cdots & \alpha_{i_t} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{i_1}^{t-1} & \alpha_{i_2}^{t-1} & \cdots & \alpha_{i_t}^{t-1} \end{bmatrix} \begin{bmatrix} \frac{1}{g_{i_1(\alpha_{i_1})}} & & & \\ & \frac{1}{g_{i_2(\alpha_{i_2})}} & & \\ & & ... & \\ & & & \frac{1}{g_{i_t(\alpha_{i_t})}} \end{bmatrix} \begin{bmatrix} e_{i_1} \\ e_{i_2} \\ \vdots \\ e_{i_t} \end{bmatrix}.$$

2. After obtaining the correct codeword, we also need to decode the message symbols $m_1, \ldots, m_k$. A naive method is to solve a system of $k \times k$ system of linear equations. Since RS code is MDS, we can arbitrarily pick k coded symbols and solve for the k message symbols. This requires $O(k^3)$ steps. Show that the following procedure can also produce the message symbols.

Input: a valid codeword $\mathbf{c} = (c_1, c_2, \ldots, c_n)$ in the row-space of matrix G in (5).
Output: a message vector $(m_1, \ldots, m_k)$ such that $(m_1, \ldots, m_k) \cdot G = \mathbf{c}$.
Step 0. Let $\mathbf{x} \leftarrow \mathbf{c}$.
Step 1. $\ell \leftarrow k$.
Step 2. Compute the ℓ-th message symbol m_ℓ by taking the inner product

$$m_\ell \leftarrow \mathbf{x} \cdot \big(\frac{\alpha_1^{n-\ell}}{g_1(\alpha_1)}, \frac{\alpha_2^{n-\ell}}{g_2(\alpha_2)}, \ldots, \frac{\alpha_n^{n-\ell}}{g_n(\alpha_n)}\big).$$

Step 3. $\mathbf{x} \leftarrow \mathbf{x} - m_\ell(\alpha_1^{\ell-1}, \alpha_2^{\ell-1}, \ldots, \alpha_n^{\ell-1})$.
Step 4. $\ell \leftarrow \ell - 1$.
Step 5. While $\ell \geq 1$, go back to step 2, otherwise return $(m_1, \ldots, m_k)$ and stop.

This method computes the message symbols in the order of m_k, $m_{k-1}, \ldots, m_1$. The while-loop is repeated k times, and we need to perform $O(n)$ field operations in steps 2 and 3. The overall computational complexity is $O(kn)$.

References

[1] J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inf. Theory,* vol. 15, no. 122–127, pp., Jan. 1969.

[2] E. Berlekamp, *Algebraic coding theory,* revised edition, World Scientific Publishing, 2015.