CS 812
Lecture 33
Monday 4/13/20

Topic du jour: The Berlekamp-Massey algorithm

This is an extremely efficient method for determining the linear feedback shift register that produces a given sequence. It also plays a role in decoding of certain error-correcting codes.

Reference: J. Massey, IEEE Trans. Inform. Theory, 1969.

Connection polynomials

For this lecture we will change our point of view somewhat and consider a variation of the characteristic polynomial we have defined.

Suppose a 0-1 sequence x_i satisfies the recurrence relation (in $\mathbf{Z}_2$)

$$x_n + c_1 x_{n-1} + \ldots + c_k x_{n-k} = 0$$

for all $n \geq k$. Then we call

$$f = 1 + c_1 Z + \cdots + c_{k-1} Z^{k-1} + c_k Z^k$$

a *connection polynomial* for the sequence. Note that if ϕ is the characteristic polynomial for a Fibonacci shift register then the reverse of ϕ is a connection polynomial.

It is important to understand that a connection polynomial can have $c_k = 0$. We will refer to k as the *length* of f.

Suppose f is a polynomial as above. We say it is correct at position n if either

$$x_n + c_1 x_{n-1} + \ldots + c_k x_{n-k} = 0,$$

or if $n < k$. We will call f correct before n if it is correct at positions $0, 1, \ldots, n-1$.

The discrepancy of f at position n is

$$x_n + c_1 x_{n-1} + \ldots + c_k x_{n-k}.$$

When $n < k$ we will also say that the discrepancy is 0.

Splicing connection polynomials

Suppose we have a guess for the connection polynomial of some LFSR. We can use it to predict sequence values until it fails, at which time we should replace it by another polynomial. This is easy to do if we have remembered a previous polynomial that we have used for this purpose.

Splicing Theorem. Suppose f is correct before, but not at, position n, and g is correct before, but not at, position m, with $m < n$. Then the polynomial

$$f + Z^{n-m}g$$

is correct for all positions $\leq n$.

Proof: Consider the discrepancy of $f + Z^{n-m}g$. This is zero before position n (because f works before position n and g works before position m), and since the discrepancies for f and g cancel, it is also zero at position n.

We will now work with two connection polynomials f and g. The first represents our best guess for the connection polynomial of the sequence, and the second represents a previous guess.

We'll think of these as anchored at particular points of the sequence.

g is anchored at the last place it failed to make a correct prediction. To get started, we take $g = 1$, anchored at position -1 of the sequence.

f also starts out at position -1. The basic step of the algorithm slides f forward one position, and computes the discrepancy between f's prediction and the sequence value at that position. If the discrepancy is zero, we do nothing and continue. If the discrepancy is 1, we apply the splicing theorem to f and g to make a new f, which is good for predicting the sequence up to and including the current position.

The trick to the algorithm is to find a good strategy for updating the previous guess g.

The basic idea is to keep g's trailing edge as far along as possible. Intuitively, this will keep the degree of f small.

Suppose we have applied the theorem. There are two cases.

If the length of f has increased, we should reset g to be the old value of f.

If the length of f has not increased, we should keep g the same.

Suppose that f, before the splicing step, has length F. In his paper, Massey proves that the length of f increases iff $\max(F, n + 1 - F) > F$, that is, iff $n \geq 2F$. We can use this as a test, rather than keeping track of the lengths of f and g.

Massey also proves (we won't do it here) that this algorithm finds the minimal-degree connection polynomial possible at every stage.

Implementing the Berlekamp-Massey algorithm

Here is pseudocode for the procedure we outlined above.

To get started, let

$$f = 1, \qquad F = 0, \qquad n = -1.$$

This means that our first connection polynomial is 1, its length (formal degree) is 0, and it is anchored at position -1. Similarly, for the previous connection polynomial, we define

$$g = 1, \qquad G = 0, \qquad m = -1.$$

We also set $H = -1$ initially.

Now we make a loop that reads bits ad infinitum:

repeat (forever)
    $n := n + 1$
    if f has nonzero discrepancy at at n:
        $h := f + Z^{n-m} \cdot g$
        $H := \max(H, G + n - m)$.
        if $(H > F)$ [length increased]:
            $g := f$
            $G := F$
            $m := n$
        $f := h$
        $F := H$

A Prediction Example Using Berlekamp-Massey

Consider the sequence

$$11110000111100001111\ldots\ .$$

Note that this repeats an 8-bit pattern that consists of 4 1's and then 4 0's.

Reviewing our notation, n is the current position, $f = 1+c_0Z+c_1Z^2+\cdots$ is the current connection polynomial, g the "old f", m the last position where g was updated, and

$$\Delta_n = x_n + c_0x_{n-1} + c_1x_{n-2} + \cdots$$

is the discrepancy at n.

Remember that we update f to be $f+Z^{n-m}g$ if $\Delta_n = 1$. Let k be the current highest degree of f. Then, g is updated to be the old f iff the new f has degree higher than k. At each step, f is a degree k connection polynomial that corresponds to the length k linear recurrence $x_n = c_0x_{n-1} + \ldots c_{k-1}x_{n-k}$.

Initial values: $f, g = 1, m = -1$.

$n = 0$	$\Delta_0 = x_0 = 1$	$f_{\text{new}} = 1 + Z^{0-(-1)} \cdot 1 = Z + 1$ $g = 1,\ m = 0,\ k = 0$
$n = 1$	$\Delta_1 = x_0 + x_1 = 0$	
$n = 2$	$\Delta_2 = x_1 + x_2 = 0$	
$n = 3$	$\Delta_3 = x_2 + x_3 = 0$	
$n = 4$	$\Delta_4 = x_3 + x_4 = 0$	$f_{\text{new}} = 1 + Z + Z^{4-0}(1) = 1 + Z + Z^4$ $g = Z + 1,\ m = 4,\ k = 4$
$n = 5$	$\Delta_5 = x_5 + x_4 + x_1 = 1$	$f_{\text{new}} = 1 + Z + Z^4 + Z^{5-4}(1 + Z) = 1 + Z^2 + Z^4$ g, m, k same
$n = 6$	$\Delta_6 = x_6 + x_4 + x_2 = 1$	$f_{\text{new}} = 1 + Z^2 + Z^4 + Z^{6-4}(1 + Z) = 1 + Z^3 + Z^4$ g, m, k same
$n = 7$	$\Delta_7 = x_7 + x_4 + x_3 = 1$	$f_{\text{new}} = 1 + Z^3 + Z^4 + Z^{7-4}(1 + Z) = 1$ g, m, k same
$n = 8$	$\Delta_8 = x_8 = 1$	$f_{\text{new}} = 1 + Z^{8-4}(1 + Z) = 1 + Z^4 + Z^5$ $g = 1,\ m = 8,\ k = 5$
$n = 9$	$\Delta_9 = x_9 + x_5 + x_4 = 1$	$f_{\text{new}} = 1 + Z^4 + Z^5 + Z^{9-8}(1) = 1 + Z + Z^4 + Z^5$

At this point, this connection polynomial gives us a recurrence that works for the rest of the sequence.

Prediction Using Linear Algebra

There are four consecutive 0's in the sequence, so it could not have been generated by any LFSR of length shorter than 5.

The recurrence relation for a 5-cell Fibonacci shift register is

$$x_n = c_4 x_{n-1} + c_3 x_{n-2} + c_2 x_{n-3} + c_1 x_{n-4} + c_0 x_{n-5},$$

where the c_i are 0-1 coefficients we must determine.

Using the given sequence, we get the following linear system.

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Note that we have sampled $2 \cdot 5 = 10$ bits to make this.

Since the matrix is invertible, there is a unique solution for the c_i's. Solving the system, we get $c_0 = 1$, $c_1 = 1$, $c_2 = 0$, $c_3 = 0$, $c_4 = 1$. This agrees with what we found before.

Additional Information

The Berlekamp-Massey algorithm can be thought of as a procedure that solves a special kind of linear system. For this reason, it has antecedents in the literature. See K. S. McCurley, Odds and Ends from Cryptology and Computational Number Theory, in Proc. Symp. Appl. Math., vol. 42, 1990, pp. 145-166.

The algorithm can be generalized to linear recurrences modulo composite numbers. See N. J. A. Sloane and J. A. Reeds, Shift-register synthesis (modulo m), SIAM J. Computing, v. 14, 1985, pp. 505-513.