Berlekamp - Welch (for general errors/corruptions)

Original message (length n): $m_1, m_2, \dots, m_n$

Degree n-1

↓ Lagrange Interpolation

$$P(i) = p_0 + p_1 i + \dots + p_{n-1} i^{n-1}$$

↓ Send n+2k points

$$m_1, m_2, \dots, m_n, m_{n+1}, \dots, m_{n+2k}$$

↓ k corruptions

received message: $r_1, r_2, \dots, r_{n+2k}$

Procedure: $E(i) = (i-e_1)(i-e_2)\cdots(i-e_k)$

Degree k

Define $Q(i) = P(i)E(i)$ Degree n+k-1

Observe $Q(i) = P(i)E(i) = r_i E(i)$

at $i = 1, 2, \dots, n+2k$

⇓

n+2k equations !!

Write

$$\begin{cases} E(i) = 1 \cdot i^k + b_{k-1} i^{k-1} + \dots + b_1 i + b_0 \\ Q(i) = a_{n+k-1} i^{n+k-1} + \dots + a_1 i + a_0 \end{cases}$$

⇓

n+2k unknowns

⇓ ⇓

Solve for coefficients of $Q(i)$ and $E(i)$

⇓

Obtain $\dfrac{Q(i)}{E(i)} = P(i) \Rightarrow$ recover $\begin{cases} P(1) = m_1 \\ \vdots \\ P(n) = m_n \end{cases}$

CS 70 Spring 2021 — Discrete Mathematics and Probability Theory — DIS 5A

1 Berlekamp-Welch Warm Up

Let $P(i)$, a polynomial applied to the input i, be the original encoded polynomial before sent, and let r_i be the received info for the input i which may or may not be corrupted.

(a) When does $r_i = P(i)$? When does r_i not equal $P(i)$?

No corruption occurs at that location

general error (corruption) occurs at that location

(b) If you want to send a length-n message, what should the degree of $P(x)$ be? Why?

$P(i) = p_0 + p_1 i + \cdots + p_{n-1} i^{n-1}$ $\boxed{\text{Degree } n-1}$

(c) If there are at most k erasure errors, how many packets should you send? If there are at most k general errors, how many packets should you send? (We will see the reason for this later.) Now we will only consider general errors.

Not BW! $\boxed{n+k}$ pts

Use BW

$m_1, m_2, \ldots, m_n, m_{n+1}, \ldots, m_{\boxed{n+2k}}$

(d) What do the roots of the error polynomial $E(x)$ represent? Does the receiver know the roots of $E(x)$? If there are at most k errors, what is the maximum degree of $E(x)$? Using the information about the degree of $P(x)$ and $E(x)$, what is the degree of $Q(x) = P(x)E(x)$?

No

error location

$E(i) = (i-e_1)(i-e_2)\cdots(i-e_k)$

$\boxed{\text{Degree } k}$

Define $Q(i) = P(i)E(i)$ $\boxed{\text{Degree } n+k-1}$

(e) Why is the equation $Q(i) = P(i)E(i) = r_iE(i)$ always true? (Consider what happens when $P(i) = r_i$, and what happens when $P(i)$ does not equal r_i.)

$P(i)E(i) = r_iE(i)$

Equality holds

general error occured

$E(i) = 0$

$P(i) \cdot 0 = r_i \cdot 0$

$0 = 0$ ✓

(f) In the polynomials $Q(x)$ and $E(x)$, how many total unknown coefficients are there? (These are the variables you must solve for. Think about the degree of the polynomials.) When you receive packets, how many equations do you have? Do you have enough equations to solve for all of the unknowns? (Think about the answer to the earlier question - does it make sense now why we send as many packets as we do?)

$Q(i) = P(i)E(i) = r_i E(i)$

Write

$$\begin{cases} E(i) = 1 \cdot i^k + \underbrace{b_{k-1} i^{k-1} + \cdots + b_1 i + b_0}_{k} \\ Q(i) = \underbrace{a_{n+k-1} i^{n+k-1} + \cdots + a_1 i + a_0}_{n+k} \end{cases}$$

at $i = 1, 2, \ldots, n+2k$

$\Downarrow$

$n+2k$ unknown coefficients $\Longleftrightarrow$ $n+2k$ equations !!

(g) If you have $Q(x)$ and $E(x)$, how does one recover $P(x)$? If you know $P(x)$, how can you recover the original message?

Solve for coefficients of $Q(i)$ and $E(i)$

$\Downarrow$

Obtain $\frac{Q(i)}{E(i)} = P(i) \Rightarrow$ recover

$$\left\{ \boxed{\begin{array}{l} P(1) = m_1 \\ \vdots \\ P(n) = m_n \end{array}} \right.$$

$\rightarrow$ evaluate $\rightarrow$

2 Berlekamp-Welch Algorithm

In this question we will send the message $(m_0, m_1, m_2) = (1, 1, 4)$ of length $n = 3$. We will use an error-correcting code for $k = 1$ general error, doing arithmetic over GF(5).

(a) Construct a polynomial $P(x) \pmod 5$ of degree at most 2, so that Lagrange :

$$P(0) = 1, \qquad P(1) = 1, \qquad P(2) = 4.$$

$\boxed{\begin{array}{l} P(0)=1 \\ P(1)=1 \\ P(2)=4 \end{array}}$

What is the message $(c_0, c_1, c_2, c_3, c_4)$ that is sent?

$\Delta_0 = (x-1)(x-2)\left((0-1)(0-2)\right)^{-1}$

$\Delta_1 = (x-0)(x-2)\left((1-0)(1-2)\right)^{-1}$

$\Delta_2 = (x-0)(x-1)\left((2-0)(2-1)\right)^{-1}$

$P(x)$

$= 1 \cdot \Delta_0 + 1 \cdot \Delta_1 + 4 \cdot \Delta_2$

$= 4x^2 + x + 1$

send $\rightarrow$ $\boxed{(1, 1, 4, 0, 4)}$

(b) Suppose the message is corrupted by changing c_0 to 0. Set up the system of linear equations in the Berlekamp-Welch algorithm to find $Q(x)$ and $E(x)$.

degree 3 $\quad Q(x) = a_3x^3 + a_2x^2 + a_1x + a_0 \quad (0,1,4,0,\underline{4})$

degree 1 $\quad E(x) = x + b_0 \longleftarrow x - e_0$

$$a_0 \equiv c_0(0+b_0) \equiv 0 \pmod{5}$$

$$a_3 + a_2 + a_1 + a_0 \equiv c_1(1+b_0) \equiv 1+b_0 \pmod{5}$$

$$8a_3 + 4a_2 + 2a_1 + a_0 \equiv c_2(2+b_0) \equiv 8 + 4b_0 \pmod{5}$$

$$27a_3 + 9a_2 + 3a_1 + a_0 \equiv c_3(3+b_0) \equiv 0 \pmod{5}$$

$$64a_3 + 16a_2 + 4a_1 + a_0 \equiv c_4(4+b_0) \equiv 16 + 4b_0 \pmod{5}$$

(c) Assume that after solving the equations in part (b) we get $Q(x) = 4x^3 + x^2 + x$ and $E(x) = x$. Show how to recover the original message from Q and E.

$$\begin{array}{r} \boxed{4x^2+x+1} \\ x\,\overline{)\,4x^3+x^2+x} \end{array}$$

evaluation

$$P(0)=1$$
$$P(1)=1 \quad \longrightarrow \quad \boxed{(1,1,4)}$$
$$P(2)=4$$

Problem 3, Fewer Errors

(d) Suppose you're actually trying to decode the received message $(4,4,4)$. Based on what you showed in the previous two parts, what will happen during row reduction when you try to solve for the unknowns?

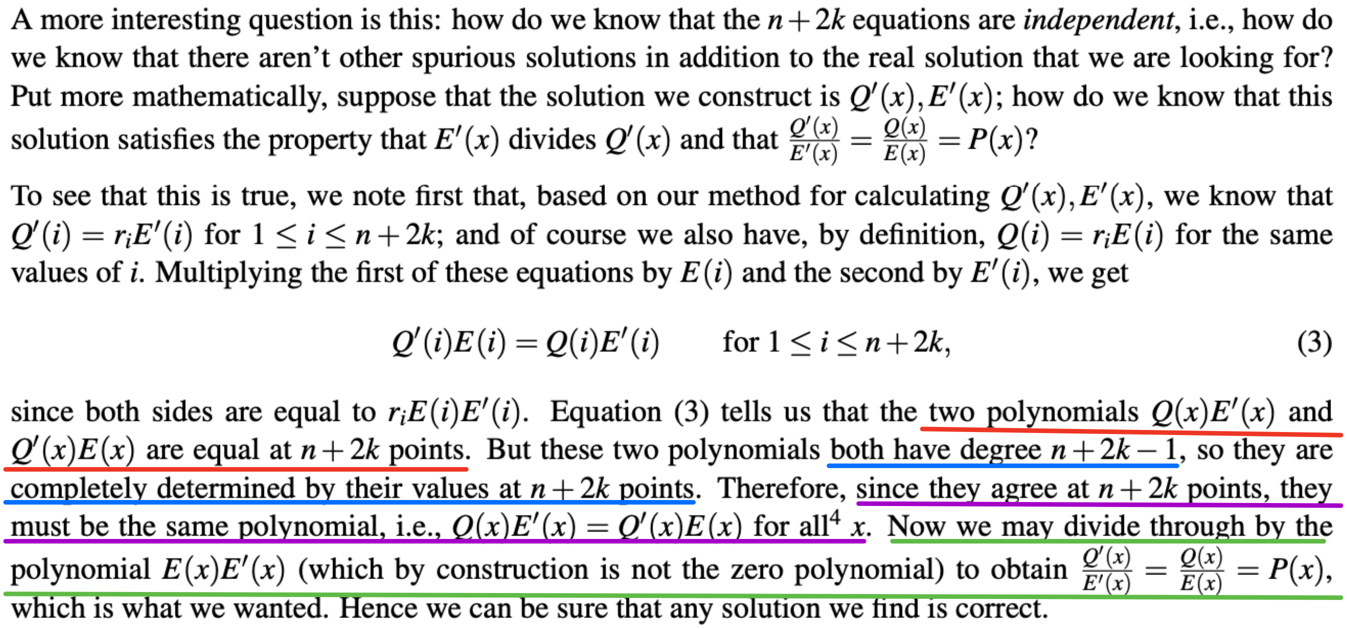
underdetermined system

↓

multiple solutions of $Q(x)$ and $E(x)$

(e) Prove that in general, no matter what the solution of $Q(x)$ and $E(x)$ are though, the recovered $P(x)$ will always be the same.

Note 9, Page 7, Section "Finer Points"



A more interesting question is this: how do we know that the $n+2k$ equations are *independent*, i.e., how do we know that there aren't other spurious solutions in addition to the real solution that we are looking for? Put more mathematically, suppose that the solution we construct is $Q'(x), E'(x)$; how do we know that this solution satisfies the property that $E'(x)$ divides $Q'(x)$ and that $\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x)$?

To see that this is true, we note first that, based on our method for calculating $Q'(x), E'(x)$, we know that $Q'(i) = r_i E'(i)$ for $1 \le i \le n+2k$; and of course we also have, by definition, $Q(i) = r_i E(i)$ for the same values of i. Multiplying the first of these equations by $E(i)$ and the second by $E'(i)$, we get

$$Q'(i)E(i) = Q(i)E'(i) \qquad \text{for } 1 \le i \le n+2k, \tag{3}$$

since both sides are equal to $r_i E(i)E'(i)$. Equation (3) tells us that the two polynomials $Q(x)E'(x)$ and $Q'(x)E(x)$ are equal at $n+2k$ points. But these two polynomials both have degree $n+2k-1$, so they are completely determined by their values at $n+2k$ points. Therefore, since they agree at $n+2k$ points, they must be the same polynomial, i.e., $Q(x)E'(x) = Q'(x)E(x)$ for all[4] x. Now we may divide through by the polynomial $E(x)E'(x)$ (which by construction is not the zero polynomial) to obtain $\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x)$, which is what we wanted. Hence we can be sure that any solution we find is correct.

3 Berlekamp-Welch Algorithm with Fewer Errors

In class we derived how the Berlekamp-Welch algorithm can be used to correct k general errors, given $n+2k$ points transmitted. In real life, it is usually difficult to determine the number of errors that will occur. What if we have less than k errors? This is a follow up to the exercise posed in the notes.

Suppose Alice wants to send 1 message to Bob and wants to guard against 1 general error. She decides to encode the message with $P(x) = 4$ (on GF(7)) such that $P(0) = 4$ is the message she want to send. She then sends $P(0), P(1), P(2) = (4,4,4)$ to Bob.

(a) Suppose Bob receives the message $(4,5,4)$. Without performing Gaussian elimination explicitly, find $E(x)$ and $Q(x)$.

(b) Now, suppose there were no general errors and Bob receives the original message $(4,4,4)$. Show that the $Q(x), E(x)$ that you found in part (a) still satisfies $Q(i) = r_i E(i)$ for all $i = 0,1,2$.

(c) Verify that $E(x) = x$, $Q(x) = 4x$ is another possible set of polynomials that satisfies $Q(i) = r_i E(i)$ for all $i = 0,1,2$.