Error Correcting Codes and Secret Sharing Intro

Secret Sharing: We make use of the fact that there is a unique polynomial of degree d passing through a given set of $d+1$ points. This means that if we require k people to come together in order to find a secret, we should use a polynomial of degree $k-1$, and give each person one point. There are more complicated schemes if there are more conditions, but they all use the same concept.

Erasure errors: A packet/point $(x, P(x))$ is *lost* in the communication channel.

1	5	3	4	3

$\longrightarrow$

1	5		4	3

Protection: (n packets, k errors) interpolate polynomial through the message points, send $n+k$ packets on the polynomial

General errors: A packet is *modified* in the communication channel.

1	5	3	4

$\longrightarrow$

1	5	6	4

Protection: (n packets, k errors) interpolate polynomial through the message points, send $n+2k$ packets on the polynomial

Berlekamp–Welch Algorithm:

Variables: sent message packets m_i, received packets r_i, error locations e_i

Polynomials:

- $P(x)$: original polynomial through message (this is what we want)
- $E(x) = (x-e_1)(x-e_2)\cdots(x-e_k)$: error locator polynomial
- $Q(x) = P(x)E(x)$, or $Q(i) = P(i)E(i) = r_iE(i)$ for all i

$Q(x)$ and $E(x)$ are unknown, but we can solve for them using a system of equations.

1 Berlekamp-Welch Warm Up

Note 8
Note 9

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

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

(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.

(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 $Q(x) = P(x)E(x)$?

(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.)

(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?)

(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?

Solution:

(a) P has degree $n-1$ since n points would determine a degree $n-1$ polynomial.

(b) $r_i = P(i)$ when the received packet is correct. r_i does not equal $P(i)$ the received packet is corrupted.

(c) We send $n+k$ packets when we have k erasures and $n+2k$ packets for k general errors.

(d) The roots of error polynomial $E(x)$ represent the locations of corrupted packets. The receiver does not know the roots of $E(x)$. $E(x)$ is a polynomial that the receiver needs to compute in order to obtain $P(x)$. If there are at most k errors, then the maximum degree of $E(x)$ is k. The maximum degree of Q is $(n-1)+(k) = n+k-1$ since the degree of P is $n-1$ and the degree of E is at most k.

(e) If there is no error at point i, $P(i) = r_i$ and then multiplying each side by $E(i)$ gives $P(i)E(i) = r_iE(i)$. If there is an error at point i, then $E(i) = 0$, which means $P(i)E(i) = r_iE(i) = 0$.

(f) The maximum degree of $Q(x)$ is $n+k-1$, so the number of unknowns is $n+k$. The maximum degree of $E(x)$ is k, which would mean there would be $k+1$ unknowns. However, we know that the coefficient of x^k is 1 in $E(x)$, so the number of unknowns is k.

The total number of unknowns is $(n+k)+(k) = n+2k$

There are $n+2k$ equations, which is enough to solve for $n+2k$ unknowns.

(g) We can compute $P(x)$ using the equation: $P(x) = Q(x)/E(x)$. To recover the message, we compute $P(i)$ for $1 \leq i \leq n$.

2 Berlekamp-Welch Algorithm

Note 8
Note 9

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

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

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

(b) Suppose you receive the message $(0, 1, 4, 0, 4)$ and know that one packet was corrupted. Set up the system of linear equations in the Berlekamp-Welch algorithm to find $Q(x)$ and $E(x)$.

(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.

Solution:

(a) We use Lagrange interpolation to construct the unique quadratic polynomial $P(x)$ such that $P(0) = m_0 = 1, P(1) = m_1 = 1, P(2) = m_2 = 4$. Doing all arithmetic over GF(5), so that i.e. $2^{-1} = 3 \pmod 5$,

$$\begin{aligned}
\Delta_0(x) &= \frac{(x-1)(x-2)}{(0-1)(0-2)} = \frac{x^2-3x+2}{2} \equiv 3(x^2-3x+2) \pmod 5 \\
\Delta_1(x) &= \frac{(x-0)(x-2)}{(1-0)(1-2)} = \frac{x^2-2x}{-1} \equiv 4(x^2-2x) \pmod 5 \\
\Delta_2(x) &= \frac{(x-0)(x-1)}{(2-0)(2-1)} = \frac{x^2-x}{2} \equiv 3(x^2-x) \pmod 5 \\
P(x) &= m_0\Delta_0(x) + m_1\Delta_1(x) + m_2\Delta_2(x) \\
&= 1\Delta_0(x) + 1\Delta_1(x) + 4\Delta_2(x) \\
&\equiv 4x^2 + x + 1 \pmod 5
\end{aligned}$$

For the final message we need to add 2 redundant points of P. Since 3 and 4 are the only points in GF(5) that we have not used yet, we compute $P(3) = 0, P(4) = 4$, and so our message is $(1, 1, 4, 0, 4)$.

(b) The message received is $(c'_0, c'_1, c'_2, c'_3, c'_4) = (0, 1, 4, 0, 4)$. Let $R(x)$ be the function such $R(i) = c'_i$ for $0 \le i < 5$. Let $E(x) = x + b_0$ be the error-locator polynomial, and $Q(x) = P(x)E(x) = a_3x^3 + a_2x^2 + a_1x + a_0$. Since $Q(i) = P(i)E(i) = R(i)E(i)$ for $0 \le i < 5$, we have the following equalities $\pmod 5$

$$\begin{aligned}
Q(0) &= 0E(0) \\
Q(1) &= 1E(1) \\
Q(2) &= 4E(2) \\
Q(3) &= 0E(3) \\
Q(4) &= 4E(4)
\end{aligned}$$

which can be rewritten as a system of linear equations

$$\begin{array}{rcrcrcrcrcl} & & & & & & a_0 & & & = & 0 \\ a_3 & + & a_2 & + & a_1 & + & a_0 & - & b_0 & = & 1 \\ 8a_3 & + & 4a_2 & + & 2a_1 & + & a_0 & - & 4b_0 & = & 8 \\ 27a_3 & + & 9a_2 & + & 3a_1 & + & a_0 & & & = & 0 \\ 64a_3 & + & 16a_2 & + & 4a_1 & + & a_0 & - & 4b_0 & = & 1 \end{array}.$$

(c) From the solution, we know

$$Q(x) = 4x^3 + x^2 + x,$$
$$E(x) = x + b_0 = x.$$

Since $Q(x) = P(x)E(x)$, the recipient can compute $P(x) = Q(x)/E(x) = 4x^2 + x + 1$, the polynomial $P(x)$ from part (a) used by the sender. The error locating polynomial $E(x)$ is degree one, so there is only one error, and as $E(x) = x = x - 0$, the corrupted bit was the first one. To correct this error we evaluate $P(0) = 1$ and combine this with the two uncorrupted bits m_1, m_2, to get the original message

$$(m_0, m_1, m_2) = (1, 1, 4).$$

3 Secrets in the United Nations

Note 8

A vault in the United Nations can be opened with a secret combination $s \in \mathbb{Z}$. In only two situations should this vault be opened: (i) all 193 member countries must agree, or (ii) at least 55 countries, plus the U.N. Secretary-General, must agree.

(a) Propose a scheme that gives private information to the Secretary-General and all 193 member countries so that the secret combination s can only be recovered under either one of the two specified conditions.

(b) The General Assembly of the UN decides to add an extra level of security: each of the 193 member countries has a delegation of 12 representatives, all of whom must agree in order for that country to help open the vault. Propose a scheme that adds this new feature. The scheme should give private information to the Secretary-General and to each representative of each country.

Solution:

(a) Create a polynomial of degree 192 and give each country one point. Give the Secretary General $193 - 55 = 138$ distinct points, so that if she collaborates with 55 countries, they will have a total of 193 points and can reconstruct the polynomial. Without the Secretary-General, the polynomial can still be recovered if all 193 countries come together. (We do all our work in GF(p) for some large prime $p \geq 1 + 193 + 138$, since we need to distribute a point to each of the countries, 138 points to the Secretary General, and one point for the secret).

Alternatively, we could have one scheme for condition (i) and another for (ii). The first condition is the secret-sharing setup we discussed in the notes, so a single polynomial of degree 192 suffices, with each country receiving one point, and evaluation at zero returning the combination s. For the second condition, create a polynomial f of degree 1 with $f(0) = s$, and give $f(1)$ to the Secretary-General. Now create a second polynomial g of degree 54, with $g(0) = f(2)$, and give one point of g to each country. This way any 55 countries can recover $g(0) = f(2)$, and then can consult with the Secretary-General to recover $s = f(0)$ from $f(1)$ and $f(2)$.

(b) We'll layer an *additional* round of secret-sharing onto the scheme from part (a). If t_i is the key given to the ith country, produce a degree-11 polynomial f_i so that $f_i(0) = t_i$, and give one point of f_i to each of the 12 delegates. Do the same for each country (using different f_i each time, of course).