CS 70 Discrete Mathematics and Probability Theory
Spring 2018 Satish Rao and Babak Ayazifar

DIS 5B

1 Error-Detecting Codes

In the realm of error-correcting codes, we usually want to recover the original message if we detect any errors, and we want to provide a guarantee of being able to do this even if there are k general errors. Suppose that instead we are satisfied with detecting whether there is any error at all and do not care about the original message if we detect any errors. In class you saw that for recovering from at most k general errors when transmitting a message of length n you need to extend your message by $2k$ symbols and send a message of length $n+2k$. But since we don't require recovering the original message, it is conceivable that we might need less symbols.

Formally, suppose that we have a message consisting of n symbols that we want to transmit. We want to be able to detect whether there is any error if we are guaranteed that there can be at most k general errors. That is, your receiver should be able to say either 'this message is completely correct' and decode it, or say 'this message has at least one error' and throw it away. How should we extend our message (i.e. by how many symbols should we extend, and how should we get those symbols) in order to be able to detect whether our message has been corrupted on its way? You may assume that we work in $GF(p)$ for a very large prime number p. Show that your scheme works, and that adding any lesser number of symbols is not good enough.

Solution: We claim that we need to extend our message by k symbols in order to be able to detect up to k errors. Suppose that the message is extended to $m_1, m_2, \ldots, m_k, m_{k+1}, \ldots, m_{k+n}$. The encoding procedure is exactly the same as what we saw in the lecture. We do the following detection algorithm. Find the unique polynomial of degree $n-1$ that passes through points $(1, m_1)$ up to (n, m_n). Call this polynomial to $P(x)$. If all of the points corresponding to the extended symbols $(n+1, m_{n+1})$ up to $(n+k, m_{n+k})$ lie on $P(x)$ then we declare that there are no errors. Otherwise the message is corrupted.

Now we argue that why the above extended message and the detection algorithm work:
We know that there are at most k errors. In other words, at least n of the points are still correct. We know that n points are enough to fully determine a degree at most $n-1$ polynomial, and since these points are all still "correct", the only polynomial of degree $n-1$ that goes through them is the original polynomial that interpolated the n original symbols. So if any of the other points have been changed, and therefore do not lie on the original polynomial, then the original polynomial would not match these points, so there would be no degree $n-1$ polynomial that matches all the points sent. Otherwise, if all the points are unchanged, it is clear that by our construction that we will get the original polynomial going through the points.

Now we prove that if we extend the message by any number of symbols less than k, then the adversary can perturb the symbols such that the detector does not find it out. Suppose that the

message is extended by $k-1$ symbols, m_{n+1} up to m_{n+k-1}. Then the adversary can change symbol m_n to $\tilde{m}_n$, and find the polynomial that passes through the points $(1, m_1)$ up to $(n, \tilde{m}_n)$. Let this polynomial be $\tilde{P}(x)$. Then the adversary can change the extended symbols such that all of the points $(n+1, \tilde{m}_{n+1})$ up to $(n+k-1, \tilde{m}_{n+k-1})$ lie on $\tilde{P}(x)$. (Note that the adversary can perturb up to k symbols) Therefore, the detector cannot detect that the message is corrupted. This shows that we need at least k extra symbols to detect k errors.

2 Berlekamp-Welch Warm Up

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

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

(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 $E(x)$ tell you? 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 $Q(i) = P(i)E(i) = r_i E(i)$ 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) The received packet is correct; the received packet is corrupted.

(b) P has degree at most $n-1$ since n points determine a degree $\leq n-1$ polynomial.

(c) $n+k$; $n+2k$.

(d) The locations of corrupted packets. No. k. The degree of Q is $(n-1)+(k) = n+k-1$.

(e) If $P(i) = r_i$, then $P(i)E(i) = r_i E(i)$. If $P(i) \neq r_i$, then $E(i) = 0$.

(f) $(n+k-1+1)+(k) = n+2k$ unknowns. There are $n+2k$ equations. Yes.

(g) $P(x) = Q(x)/E(x)$. Compute $P(i)$ for $1 \leq i \leq n$. Alternatively, since we know the error-locator polynomial $E(x)$, we can find its roots to figure out which packets were corrupted and then we only need to evaluate $P(x)$ at the locations of the errors.

3 Berlekamp-Welch for General Errors

Suppose that Hector wants to send you a length $n = 3$ message, m_0, m_1, m_2, with the possibility for $k = 1$ error. For all parts of this problem, we will work mod 11, so we can encode 11 letters as shown below:

A	B	C	D	E	F	G	H	I	J	K
0	1	2	3	4	5	6	7	8	9	10

Hector encodes the message by finding the degree ≤ 2 polynomial $P(x)$ that passes through $(0, m_0)$, $(1, m_1)$, and $(2, m_2)$, and then sends you the five packets $P(0), P(1), P(2), P(3), P(4)$ over a noisy channel. The message you receive is

$$\text{DHACK} \Rightarrow 3, 7, 0, 2, 10 = r_0, r_1, r_2, r_3, r_4$$

which could have up to 1 error.

(a) First, let's locate the error, using an error-locating polynomial $E(x)$. Let $Q(x) = P(x)E(x)$. Recall that

$$Q(i) = P(i)E(i) = r_i E(i), \quad \text{for} \quad 0 \leq i < n + 2k.$$

What is the degree of $E(x)$? What is the degree of $Q(x)$? Using the relation above, write out the form of $E(x)$ and $Q(x)$ in terms of the unknown coefficients, and then a system of equations to find both these polynomials.

(b) Solve for $Q(x)$ and $E(x)$. Where is the error located?

(c) Finally, what is $P(x)$? Use $P(x)$ to determine the original message that Hector wanted to send.

Hint: The message refers to a US federal agency.

Solution:

(a) The degree of $E(x)$ will be 1, since there is at most 1 error. The degree of $Q(x)$ will be 3, since $P(x)$ is of degree 2. $E(x)$ will have the form $E(x) = x + e$, and $Q(x)$ will have the form $Q(x) = ax^3 + bx^2 + cx + d$. We can write out a system of equations to solve for these 5 variables:

$$\begin{aligned} d &= 3(0+e) \\ a+b+c+d &= 7(1+e) \\ 8a+4b+2c+d &= 0(2+e) \\ 27a+9b+3c+d &= 2(3+e) \\ 64a+16b+4c+d &= 10(4+e) \end{aligned}$$

Since we are working mod 11, this is equivalent to:

$$\begin{aligned} d &= 3e \\ a+b+c+d &= 7+7e \\ 8a+4b+2c+d &= 0 \\ 5a+9b+3c+d &= 6+2e \\ 9a+5b+4c+d &= 7+10e \end{aligned}$$

(b) Solving this system of linear equations we get

$$Q(x) = 3x^3 + 6x^2 + 5x + 8.$$

Plugging this into the first equation (for example), we see that:

$$d = 8 = 3e \quad \Rightarrow \quad e = 8 \cdot 4 = 32 \equiv 10 \bmod 11$$

This means that

$$E(x) = x + 10 \equiv x - 1 \mod 11.$$

Therefore, the error occurred at $x = 1$ (so the second number sent in this case).

(c) Using polynomial division, we divide $Q(x) = 3x^3 + 6x^2 + 5x + 8$ by $E(x) = x - 1$:

$$P(x) = 3x^2 + 9x + 3$$

Then, $P(1) = 3 + 9 + 3 = 15 \equiv 4 \bmod 11$. This means that our original message was

$$3, 4, 0 \quad \Rightarrow \quad \text{DEA}.$$

Note: In Season 4 of Breaking Bad, Hector Salamanca (who cannot speak), uses a bell to spell out "DEA" (Drug Enforcement Agency).