Lecture 3C:
Error Correction

UC Berkeley EECS 70
Summer 2022
Tarang Srivastava

Announcements!

- Read the Weekly Post
- **HW 3** and **Vitamin 3** have been released, due **Today** (grace period Fri)
- Tarang's Last Lecture, Michael will begin starting next week
- Midterm is 7/15 (6-8p)
- Midterm Scope
  - Notes: 1-11
  - HW: 1-4
  - Lectures: 1A-4B
  - Discussions: 1A-4B
  - Topics: Up to and including countability. (Computability will not be on the midterm)
- Midterm format will be different from previous semesters. More proofs.

Review

Property 1: A non-zero polynomial of degree d has at most d roots
Property 2: Any d+1 points define a unique degree d polynomial] main idea secret sharing

Claim 2: A polynomial of degree d with roots a_1, …, a_k can be written as $p(x) = c(x-a_1)...(x-a_k)$.

$x^2 - 2x + 1$
$(x-1)(x-1)$

<u>From Discussion 3B:</u>
if f and g are degree x and degree y then
- $f + g$ is at most degree $max(x, y)$
- $f \cdot g$ is at most degree $x + y$
- f / g is at most degree $x - y$

Review (cont.)

Secret Sharing:

Problem: We need any k out of n people to agree to unlock some code.

Solution:

1. Create a degree k-1 polynomial $p(x)$
2. Encode the secret in the polynomial ($p(0)$ = "*secret*").
3. Give a point that the polynomial contains to each person (generate n points)
4. Any k points can be used to reconstruct the degree k-1 polynomial $p(x)$

$n > k$

Review of Gaussian Elimination

Why do d+1 points define a degree d polynomial uniquely?

A degree d polynomial has d + 1 coefficients:

d+1 coefficients

$$f(x) = a_d x^d + a_{d-1} x^{d-1} + \ldots + a_2 x^2 + a_1 x + a_0 \pmod{p}$$

So, we need d + 1 equations to solve for d + 1 unknowns.
We get d + 1 equations by plugging in the d + 1 points.

degree 3: $a_3 x^3 + a_2 x^2 + a_1 x + a_0$

$x \quad p(x)$

$0 \rightarrow 1$

$a_3 (0)^3 + a_2 (0)^2 + a_1 \cdot 0 + a_0 = 1$

Erasure Errors

Send some message across an **unreliable** channel.
The channel randomly **drops** k packets.

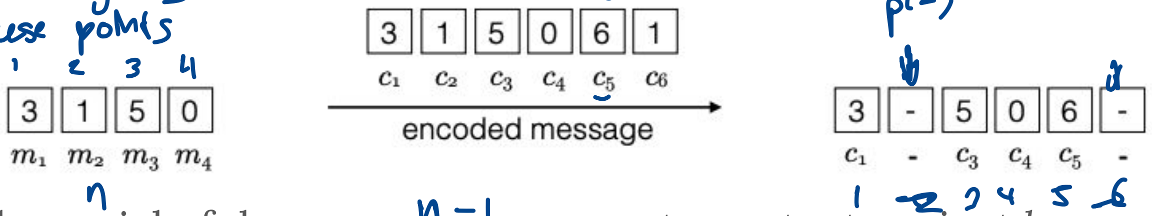
sender reciever

How can we **recover** our original message? Polynomials!

We want to encode our message into a polynomial, and then generate k extra packets. Then with any n received packets we can reconstruct the polynomial and get the original message.

p(x) that goes through these points

n+4 p(5) p(6) p(2)



Construct a polynomial of degree __n-1_______ to protect against k erasures.

Bob sends message with erasure protection

Bob wants to send the message "3 1 5 0" to Alice.

Bob knows that at most 2 packets will drop when sending the message to Alice.

n := message length (4) *k := maximum erasures* (2)

Message "3 1 5 0" become points "(1, 3)" "(2, 1)" "(3, 5)" "(4, 0)"

Find a degree 3 polynomial that goes through these points in GF(7)

index 1 2 3 4

value $\boxed{3}\ \boxed{1}\ \boxed{5}\ \boxed{0}$

$\tilde{m}_1$ $\tilde{m}_2$ $\tilde{m}_3$ $\tilde{m}_4$

points (index, value)

1) interpolation 2) Gaussian Elimination

$p(x) = ax^3 + bx^2 + cx + d$

$3 = a(1)^3 + b(1)^2 + c(1) + d$

$1 = a(2)^3 + b(2)^2 + c(2) + d$

$\vdots$

$a + b + c + d = 3$

$a + 4b + 2c + d = 1$

$6a + 2b + 3c + d = 5$

$a + 2b + 4c + d = 0$

$a = 1$

$b = 4$

$c = 0$

$d = 5$

$p(x) = x^3 + 4x^2 + 5$

What are the extra points Bob generates?

$p(5) = 5^3 + 4(5)^2 + 5 = 6$ (5, 6)

$p(6) = 6^3 + 4(6)^2 + 5 = 1$ (6, 1)

Bob Sends $\boxed{3\ 1\ 5\ 0\ 6\ 1}$

Alice receives message with erasure errors

$\boxed{3}\ \boxed{-}\ \boxed{5}\ \boxed{0}\ \boxed{6}\ \boxed{1}$

GF(7)

Alice receives the points (1, 3); (3, 5); (4, 0); (5, 6). How can Alice reconstruct the polynomial?

$p(x) = ax^3 + bx^2 + cx + d$

$$3 = a + b + c + d$$
$$5 = 6a + 2b + 3c + d$$
$$0 = a + 2b + 4c + d$$
$$6 = 6a + 4b + 5c + d$$

$\rightsquigarrow$ $a = 1$, $b = 4$, $c = 0$, $d = 5$

$p(x) = x^3 + 4x^2 + 5$

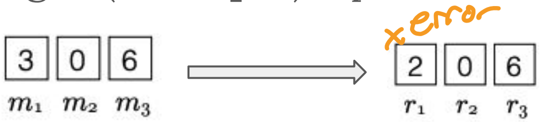
$p(2) = 1$

$\underbrace{3\ 1\ 5\ 0}$

What if you drop less than k packets?

General Errors

Send some message across a **noisy** channel.
The channel randomly changes (**corrupts**) k packets



How can we **recover** our original message?

$e(x) = (x-1)$ indicates error at index 1

This is much harder that Erasure Errors because…
1. locate where the error occurs
2. recover the correct value

Erasure Errors: Send $n + k$ packets to protect against k erasures
General Errors: Send $n + 2k$ packets to protect against k **corruptions**.

Solution: Berlekamp-Welch

Message: $m_1, \ldots, m_n$ (length = n)

Sender:

1. Form degree n-1 polynomial $p(x)$ where $p(i) = m_i$
2. Send $p(1), \ldots, p(n + 2k)$

Same as erasure

Receiver:

could be corrupted

1. Receive $r_1, \ldots, r_{n+2k}$
2. Solve $n + 2k$ equations, $q(i) = e(i)\, r_i$ to find $q(x) = e(x)p(x)$ and $e(x)$
3. Compute $p(x) = q(x)/e(x)$
4. Compute $p(1), \ldots, p(n)$ to get original message

Here r_i are the received points possibly with errors.
$p(x)$ is the original polynomial the sender used, receiver doesn't know yet
$e(x)$ is an error locator polynomial. $e(x) = (x-e_1)\ldots(x-e_k)$ where e_i is the index where the error occurs
$e(x) = 0$ when you plug in a x value where error occurs. Receiver doesn't know $e(x)$ yet.
$q(x) = e(x)p(x)$. So, we find $q(x)$ and $e(x)$ to get $p(x)$.

Berlekamp-Welch (cont.)

$q(1) = e(1)p(1) = e(1) \cdot r_1$

$q(2) = e(2)p(2) = e(2)\, r_2$

$\vdots$

$q(n+2k) = e(n+2k)\,p(n+2k) = e(n+2k)\, r_{n+2k}$

Receiver:

1. Receive $r_1, \ldots, r_{n+2k}$
2. Solve $n + 2k$ equations, $q(i) = e(i)p(i) = e(i)\, r_i$ to find $q(x) = e(x)p(x)$ and $e(x)$ is error locator polynomial. $e(i) = 0$ when there is an error in index i
3. Compute $p(x) = q(x)/e(x)$
4. Compute $p(1), \ldots, p(n)$ to get original message

leading coefficient will be 1

$e(x) = (x-e_1)\cdot(x-e_2)\cdots(x-e_k)$

$= x^k + \cdots$

$q(x) = p(x)\cdot e(x)$

$\deg p(x) = n-1$

$\deg e(x) = k$

Case 1: $p(i) = r_i$

$e(i)p(i) = e(i)\, r_i$

Case 2: $p(i) \neq r_i$

$e(i)p(i) = e(i)\, r_i$

$e(i) = 0$

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

$0 = 0$ ✓

coefficients

What is the degree of $q(x)$? $n+k-1$ How many unknowns? $n+k$

What is the degree of $e(x)$? k How many unknowns? ~~$k+1$~~ k

We have $n+2k$ unknowns in total and $n+2k$ equations

Bob sends message with corruption protection

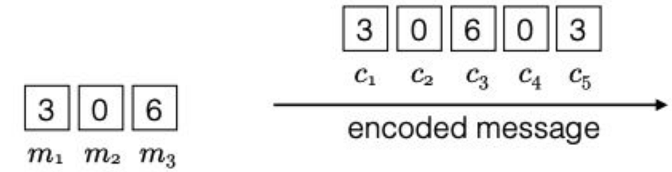
Bob wants to send the message “3 0 6” to Alice.
Bob knows that at most 1 packet will be **corrupted** when sending the message to Alice.
n := message length (3) *k := maximum corruptions* (1) $n+2k = 5$
Find a degree 2 polynomial that goes through these points in GF(7)

$p(x) = x^2 + x + 1$ ← from points (1,3) (2,0) (3,6)

$p(4) = 0$

$p(5) = 3$

extra



What are the extra points Bob generates?

(4,0)

(5,3)

Alice receives message with corruption errors

GF(7)

✓ fix 3

2	0	6	0	3
r_1	r_2	r_3	r_4	r_5

How can Alice find where the error is and fix it?

$q(x) = p(x)\,e(x)$

$\deg q := 3$

$\deg e := 1$

$q(x) = a_3x^3 + a_2x^2 + a_1x + a_0$

$e(x) = x + b_0$

i	$q(i) = p(i)e(i) = r_i e(i)$
1	$a_3(1)^3 + a_2(1)^2 + a_1(1) + a_0 = 2(1+b_0)$
2	$a_3(2)^3 + a_2(2)^2 + a_1(2) + a_0 = 0(2+b_0)$
3	$\vdots$
$\vdots$	
$n+2k$	

$a_3 + a_2 + a_1 + a_0 - 2b_0 = 2$, where $-2b_0 \equiv 5b_0$

$$a_3 + a_2 + a_1 + a_0 + 5b_0 = 2$$
$$a_3 + 4a_2 + 2a_1 + a_0 = 0$$
$$6a_3 + 2a_2 + 3a_1 + a_0 + b_0 = 4$$
$$a_3 + 2a_2 + 4a_1 + a_0 = 0$$
$$6a_3 + 4a_2 + 5a_1 + a_0 + 4b_0 = 1$$

$a_3 = 1$, $a_2 = 0$, $a_1 = 0$, $a_0 = 6$ → $q(x) = (1)x^3 + (0)x^2 + (0)x + 6 = x^3 + 6$

$b_0 = 6$ → $e(x) = x + 6 \equiv (x-1)$

error at index 1

$p(x) = \dfrac{q(x)}{e(x)}$

$$(x^3 + 0 + 0 + 6) \div (x-1) = x^2 + x + 1$$

$p(x) = x^2 + x + 1$

correct value $= p(1)$

3 ✓

Alice receives same message with NO corruption errors

3	0	6	0	3

Will Alice still get the same correct answer?

$Q(x)$ and $E(x)$ are the same

$p(x)$ is unique from Berlekamp-Welch

Thm: Any solution to Berlekamp-Welch will result in the same final $p(x)$

Proof:

Assume there's another solution $Q'(x)$ and $E'(x)$
they satisfy

$$Q'(i) = r_i E'(i) \qquad 1 \le i \le n+2k$$

$$Q'(i)\, \cancel{r_i} E(i) = r_i E'(i)\, r_i E(i) = \cancel{r_i} E'(i) \cdot Q(i)$$

$$\frac{Q'(i)\, \cancel{E(i)}}{E'(i)\, \cancel{E(i)}} = \frac{\cancel{E'(i)} \cdot Q(i)}{\cancel{E'(i)}\, E(i)} \qquad 1 \le i \le n+2k$$

$$\frac{Q'(i)}{E'(i)} = \frac{Q(i)}{E(i)} = P(i)$$

$p(x)$ is unique from Berlekamp-Welch

Thm: Any solution to Berlekamp-Welch will result in the same final $p(x)$

Proof: