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

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

<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 $p(x)$

Review of Gaussian Elimination

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

A degree d polynomial has $d + 1$ coefficients:

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

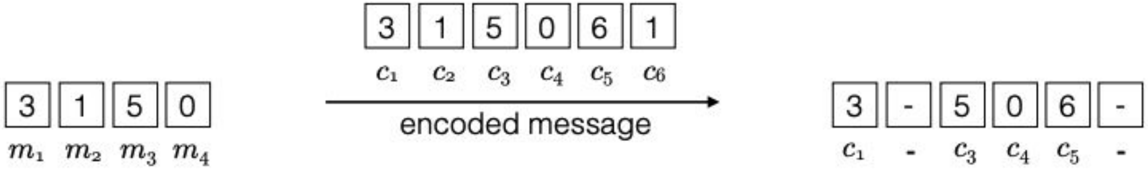
So, we need $d + 1$ equations to solve for $d + 1$ unknowns.
We get $d + 1$ equations by plugging in the $d + 1$ points.

Erasure Errors

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

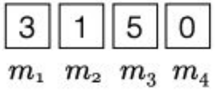
How can we **recover** our original message? Polynomials!

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.



Construct a polynomial of degree ___________ 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)

What are the extra points Bob generates?

Alice receives message with erasure errors

3	-	5	0	6	-

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

General Errors

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

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

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 $p(1), \ldots, p(n + 2k)$

Receiver:

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.)

<u>Receiver:</u>

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

What is the degree of $q(x)$?________ How many unknowns? _________

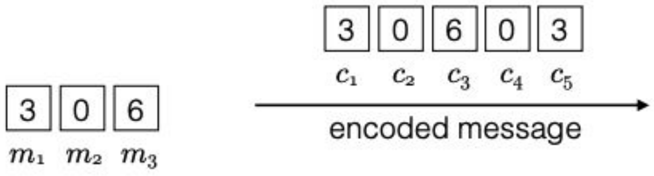
What is the degree of $e(x)$? _________ How many unknowns?_______

We have __________ unknowns in total and _________ 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)
Find a degree 2 polynomial that goes through these points in GF(7)

What are the extra points Bob generates?



Alice receives message with corruption errors

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?

Alice receives same message with NO corruption errors

3	0	6	0	3

Will Alice still get the same correct answer?

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

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

Proof: