

Error Correcting Codes

CS70: Discrete Mathematics and Probability Theory

UC Berkeley – Summer 2026

Lecture 11

Ref: Note 9

Today

Last time:

- Shared (and sort of kept) secrets

Today: Dealing with errors

- Tolerate (identified) loss: erasure codes

- Tolerate (unidentified) corruption: error correcting codes
 - ... using a beautiful decoding algorithm

Review: Interpolation via Linear Equations

Problem: Find coefficients for $P(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \cdots + a_1x + a_0$ going through points $(x_1, y_1); (x_2, y_2) \cdots (x_k, y_k)$.

... k points gives degree (at most) $k - 1$ polynomial – working mod p :

$$a_{k-1}x_1^{k-1} + \cdots + a_0 \equiv y_1 \pmod{p}$$

$$a_{k-1}x_2^{k-1} + \cdots + a_0 \equiv y_2 \pmod{p}$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$a_{k-1}x_k^{k-1} + \cdots + a_0 \equiv y_k \pmod{p}$$

Will this always work? Yes!

Linear algebra language: Powers of different x are linearly independent...

This is a Vandermonde matrix.

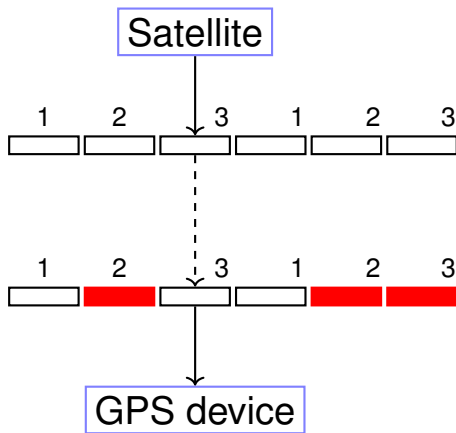
Also follows from polynomial properties:

Modular Arithmetic Fact: Exactly 1 polynomial of degree $\leq d$ with arithmetic modulo prime p contains $d + 1$ pts.

Another Uses of Polynomials! Erasure Codes

Problem: Satellite communication is unreliable – may lose packets.

⇒ *We want to get the data even if some packets are lost (erased)*



3 packet message. So send 6!

Lose 3 out 6 packets.

Gets packets 1,1,and 3.

Exploring the Problem

Problem parameters: n packet message, channel that loses up to k packets.

“Can’t get something for nothing theorem” (information theory version):

Can’t send n packets of information in $< n$ packets

$\Rightarrow$ If we might lose k packets, must send $\geq n + k$ packets

We want: Any n packets should allow reconstruction of n packet message.

Where have we seen something like this.....

Any n point values allow reconstruction of degree $n - 1$ polynomial.

Surely that’s not just a coincidence, is it?

(Hint: If it was, I wouldn’t be standing here talking about it...)

The Scheme

Problem: Want to send a message with n packets.

Channel: Lossy channel: loses k packets.

Question: Can you send $n + k$ packets and recover message?

Core idea: A degree $n - 1$ polynomial determined by any n points!

Erasure Coding Scheme: message = $m_1, m_2, \dots, m_n$ – each b bits

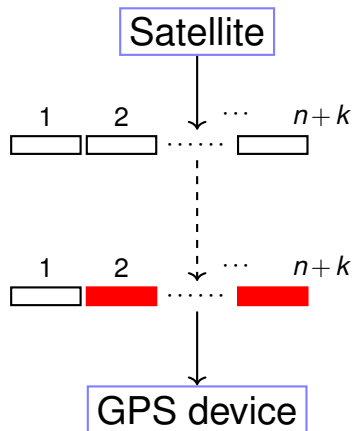
- 1 Choose prime p a little larger than $\max(2^b, n + k)$
- 2 Find interpolating polynomial of $(1, m_1), (2, m_2), \dots, (n, m_n)$
$$P(x) = a_{n-1}x^{n-1} + \dots + a_0 \pmod{p}$$
- 3 Send $(1, P(1)), (2, P(2)), \dots, (n + k, P(n + k))$

Any n of the $n + k$ packets gives polynomial

With polynomial, compute $P(1), P(2), \dots, P(n)$ – the message!

Alternative: Message packets are coefficients – efficient, but less symmetric

Erasure Codes – Summary



n packet message. So send $n + k$!

Lose k packets.

Any n packets is enough!

n packet message received

Must send $n + k$ packets $\Rightarrow$ Optimal!

Transmission Efficiency

How large a p do we need? Same basic issue as in secret sharing.

Using prime p – can encode p values, so need $p \geq 2^b$ (prime so $> 2^b$)

Can choose a prime between 2^b and 2^{b+1}

Larger than needed, but “excess” is 1 bit per packet

Also need to label packets, so you know which make it through

Math Magic: There are Galois Fields $GF(2^b)$ that “fit exactly”

Also need enough points for evaluation at different x (so $\geq n + k$)

$\Rightarrow$ Prime $p > \max(2^b, n + k)$

Information content comparison:

Secret Sharing: each share is size of whole secret

Erasur Coding: Each packet has size $1/n$ of the whole message

Computation time:

Sender: Interpolation, evaluation

Receiver: Interpolation, evaluation

No worse than $O(n^2)$ field operations (and better algorithms!)

Erasure Code: Example

Want to send 3-packet message $\langle 1, 4, 4 \rangle$

Need a polynomial through $P(1) = 1$, $P(2) = 4$, $P(3) = 4$

Interpolation... How?

- Lagrange Interpolation

- Linear System

Parameters:

- Small messages (fit in $GF(5)$)

- $n = 3$ (length of message)

- $k = 3$ (possible packets lost)

Working over $GF(p)$ — need p big enough for packets, and $p \geq n + k$

What should we use?

Example: Sender's Computation

Need a polynomial through $P(1) = 1$, $P(2) = 4$, $P(3) = 4$

Linear equations:

$$P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7} \quad (L1)$$

$$P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \pmod{7} \quad (L2)$$

$$P(3) = 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7} \quad (L3)$$

Eliminate a_2 from $L2$ (as $L2 - 4 \cdot L1 \rightarrow L2'$): $5a_1 + 4a_0 \equiv 0 \pmod{7} \quad (L2')$

Eliminate a_2 from $L3$ (as $L3 - 2 \cdot L1 \rightarrow L3'$): $a_1 + 6a_0 \equiv 2 \pmod{7} \quad (L3')$

Eliminate a_1 from $L3'$ (as $L2' - 5 \cdot L3' \rightarrow L3''$): $2a_0 \equiv 4 \pmod{7} \quad (L3'')$

Use $L3''$ to solve for a_0 : $a_0 \equiv 2 \pmod{7}$ (mult inv of 2 (mod 7) is 4)

Use $L3'$ to solve for a_1 : $a_1 \equiv 4 \pmod{7}$

Use $L1$ to solve for a_2 : $a_2 \equiv 2 \pmod{7}$

$$P(x) = 2x^2 + 4x + 2$$

$$P(1) = 1, P(2) = 4, \text{ and } P(3) = 4 \quad \text{and} \quad P(4) = 1, P(5) = 2, \text{ and } P(6) = 0$$

Send packets: $(1, 1), (2, 4), (3, 4), (4, 1), (5, 2), (6, 0)$

Example: Receiver's Computation

Sender sends: $(1, 1), (2, 4), (3, 4), (4, 1), (5, 2), (6, 0)$

Packets 2, 5, and 6 lost – receiver gets: $(1, 1), (3, 4), (4, 1)$

Reconstruct? Note: Same as initial motivating example...

Repeating packets results in two copies of packet 1 and none of packet 2

Lagrange or linear equations:

$$P(1) = a_2 + a_1 + a_0 \equiv 1 \pmod{7}$$

$$P(3) = 2a_2 + 3a_1 + a_0 \equiv 4 \pmod{7}$$

$$P(4) = 2a_2 + 4a_1 + a_0 \equiv 1 \pmod{7}$$

Solving linear equations (the magic happens...): $a_2 = 2$, $a_1 = 4$, and $a_0 = 2$

$$P(x) = 2x^2 + 4x + 2$$

Message? Evaluate! $P(1) = 1$, $P(2) = 4$, $P(3) = 4$

$\Rightarrow$ Message is $\langle 1, 4, 4 \rangle$

A Harder Problem...

Erasure Codes:

Might completely lose packets

We know when they're missing

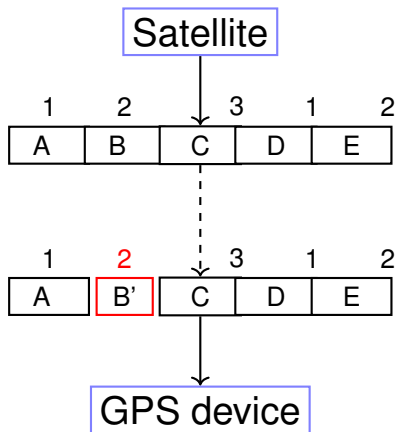
... and **which ones are missing**

Error Correction:

Noisy Channel: **corrupts** k packets (rather than **loss**)

... and **no indication which ones are corrupted!**

Error Correction



3 packet message. **Send 5.**

Corrupts 1 packets.

The Scheme

Problem: Communicate n packets $m_1, \dots, m_n$
... on noisy channel that corrupts $\leq k$ packets

Reed-Solomon Code:

- ① Make a degree $n - 1$ polynomial $P(x)$ that encodes message
 - $P(1) = m_1, \dots, P(n) = m_n$
 - **Comment:** could encode with packets as coefficients
- ② Send $P(1), \dots, P(n + 2k)$

After noisy channel: Receive values $r_1, r_2, \dots, r_{n+2k}$

Properties:

- (1) $P(i) = r_i$ for at least $n + k$ points
- (2) $P(x)$ is the unique degree $n - 1$ polynomial that contains $\geq n + k$ received points

Properties: Proof

$P(x)$: degree $n-1$ polynomial

Send $P(1), \dots, P(n+2k)$

Receive $r_1, \dots, r_{n+2k}$

At most k i 's where $P(i) \neq r_i$.

Properties:

- (1) $P(i) = r_i$ for at least $n+k$ points i ,
- (2) $P(x)$ is unique degree $n-1$ polynomial that contains $\geq n+k$ of the received points.

Proof: (1) Easy – only k corruptions.

(2) Is $P(x)$ only solution?

Let $Q(x)$ be a *different* solution (deg $n-1$ contains (*any!*) $n+k$ points)

$$\mathcal{Q} = \{i : Q(i) = r_i\} \quad |\mathcal{Q}| \geq n+k \quad |\overline{\mathcal{Q}}| \leq k$$

$$\mathcal{P} = \{i : P(i) = r_i\} \quad |\mathcal{P}| \geq n+k \quad |\overline{\mathcal{P}}| \leq k$$

$$|\overline{\mathcal{Q}} \cup \overline{\mathcal{P}}| \leq 2k \implies |\overline{\mathcal{Q} \cap \mathcal{P}}| \leq 2k \implies |\mathcal{Q} \cap \mathcal{P}| \geq n$$

$$\implies P(i) = r_i = Q(i) \text{ on } \mathcal{Q} \cap \mathcal{P} \quad (\geq n \text{ values})$$

$$\implies Q(i) = P(i) \text{ at } n \text{ points and degree } \leq n-1 \implies Q(x) = P(x)$$



Example: Reed-Solomon

Message: $\langle 3, 0, 6 \rangle$

Reed-Solomon Code:

Interpolation gives $P(x) = x^2 + x + 1 \pmod{7}$

$P(1) = 3, P(2) = 0, P(3) = 6 \pmod{7}$

Send: $P(1) = 3, P(2) = 0, P(3) = 6$, and $P(4) = 0, P(5) = 3$

Receiver gets: $r_1 = 3, r_2 = 1, r_3 = 6, r_4 = 0, r_5 = 3$

... *2nd packet corrupted* (no indication for receiver though!)

But $n + k = 3 + 1 = 4$ points are good ($P(i) = r_i$)

Solving – The Slow Way

Brute Force!

For each subset of $n + k$ points:

Fit degree $n - 1$ polynomial, $Q(x)$, to n of them

Check if consistent with $n + k$ of the total points

If yes, output $Q(x)$

For a subset of $n + k$ “good points” ($r_i = P(i)$):

n good points, so reconstructs $P(x)$ — verifies with k other good points

All good!

For any subset of $n + k$ points:

unique degree $n - 1$ polynomial $Q(x)$ that fits $\geq n$ of them

... and where $Q(x)$ is consistent with $n + k$ points

$\implies P(x) = Q(x)$.

Reconstructs $P(x)$ and only $P(x)$!!

Example

Send: $P(1) = 3, P(2) = 0, P(3) = 6, P(4) = 0, P(5) = 3$

Receiver gets: $r_1 = 3, r_2 = 1, r_3 = 6, r_4 = 0, r_5 = 3$

Goal: Find $P(x) = p_2x^2 + p_1x + p_0$ that contains $n + k = 3 + 1 = 4$ points.

All equations...

$$p_2 + p_1 + p_0 \equiv 3 \pmod{7}$$

$$4p_2 + 2p_1 + p_0 \equiv 1 \pmod{7}$$

$$2p_2 + 3p_1 + p_0 \equiv 6 \pmod{7}$$

$$2p_2 + 4p_1 + p_0 \equiv 0 \pmod{7}$$

$$4p_2 + 5p_1 + p_0 \equiv 3 \pmod{7}$$

Assume point 1 is wrong and solve... **no consistent solution!**

Assume point 2 is wrong and solve... **consistent solution!**

With one error, only $n + 2$ error locations – for general k (location sets)?

The Problem For General k

$P(x) = p_{n-1}x^{n-1} + \cdots p_0$ and receive $r_1, r_2, \dots, r_{n+2k}$

$$p_{n-1} + \cdots p_0 \equiv r_1 \pmod{p}$$

$$p_{n-1}2^{n-1} + \cdots p_0 \equiv r_2 \pmod{p}$$

.

$$p_{n-1}i^{n-1} + \cdots p_0 \equiv r_i \pmod{p}$$

.

$$p_{n-1}(m)^{n-1} + \cdots p_0 \equiv r_m \pmod{p}$$

Error!! ... Where??? ... Brute Force!

Could be anywhere!!! ... so try everywhere

How many? $\binom{n+2k}{k}$ possibilities for k locations

Something like $(n/k)^k$... exponential in k

Can we find where the bad packets are **efficiently**?!?!?!?

Isolating The Bad Packets

$$\begin{aligned}E(1)(p_{n-1} + \cdots p_0) &\equiv r_1 E(1) \pmod{p} \\E(2)(p_{n-1} 2^{n-1} + \cdots p_0) &\equiv r_2 E(2) \pmod{p} \\&\vdots \\E(m)(p_{n-1}(m)^{n-1} + \cdots p_0) &\equiv r_{n+2k} E(m) \pmod{p}\end{aligned}$$

Idea: Multiply equation i by 0 if and only if $P(i) \neq r_i$.

Blots out error locations – makes them irrelevant!

All equations satisfied!!!!

But which equations should we multiply by 0?

We will use a polynomial!!! That we don't know. But can find!

Errors at points $e_1, \dots, e_k$ (in diagram above, $e_1 = 2$)

Error-locator polynomial: $E(x) = (x - e_1)(x - e_2) \dots (x - e_k)$

$E(x) = 0$ if and only if $x = e_j$ for some j

Multiply equations by $E(x)$ (above $E(x) = (x - 2)$)

All equations satisfied!!

Example

Receiver gets: $r_1 = 3, r_2 = 1, r_3 = 6, r_4 = 0, r_5 = 3$

Find $P(x) = p_2x^2 + p_1x + p_0$ that contains $n + k = 3 + 1 = 4$ of the points.

Set up linear equations...

$$\begin{aligned}(1 + b_0)(p_2 + p_1 + p_0) &\equiv (3)(1 + b_0) \pmod{7} \\(2 + b_0)(4p_2 + 2p_1 + p_0) &\equiv (1)(2 + b_0) \pmod{7} \\(3 + b_0)(2p_2 + 3p_1 + p_0) &\equiv (6)(3 + b_0) \pmod{7} \\(4 + b_0)(2p_2 + 4p_1 + p_0) &\equiv (0)(4 + b_0) \pmod{7} \\(5 + b_0)(4p_2 + 5p_1 + p_0) &\equiv (3)(5 + b_0) \pmod{7}\end{aligned}$$

Error-locator polynomial: $(x - 2)$

Multiply equation i by $(i - 2)$. All equations satisfied!

But don't know the error-locator polynomial!

Do know form: $(x - e)$ or $x + b_0$

In general: $(x - e_1)(x - e_2) \dots (x - e_k) \longrightarrow x^k + b_{k-1}x^{k-1} + \dots + b_0$

4 unknowns (p_0, p_1, p_2 and b_0), but **nonlinear** equations.

Nonlinear to Linear

$$\begin{aligned} E(1)(p_{n-1} + \cdots p_0) &\equiv r_1 E(1) \pmod{p} \\ &\vdots \\ E(i)(p_{n-1} i^{n-1} + \cdots p_0) &\equiv r_i E(i) \pmod{p} \\ &\vdots \\ E(m)(p_{n-1} (n+2k)^{n-1} + \cdots p_0) &\equiv r_m E(m) \pmod{p} \end{aligned}$$

$m = n + 2k$ satisfied equations, $n + k$ unknowns – **but nonlinear!**

Each polynomial on the left is $E(x)P(x)$ – what is the degree?

Let $Q(x) = E(x)P(x) = a_{n+k-1}x^{n+k-1} + \cdots a_0$

Equations:

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

... and **linear** in a_i and coefficients of $E(x)$!

But now more unknowns... how many?

Unknowns in $Q(x)$ and $E(x)$

$E(x)$ has degree k :

$$E(x) = x^k + b_{k-1}x^{k-1} \cdots b_0$$

$\implies$ Leading coefficient is 1 – remaining k coefficients are unknowns

$Q(x) = P(x)E(x)$ has degree $n+k-1$:

$$Q(x) = a_{n+k-1}x^{n+k-1} + a_{n+k-2}x^{n+k-2} + \cdots a_0$$

$\implies n+k$ coefficients are unknowns

Total number of unknown coefficients: $n+2k$

Solving for $Q(x)$ and $E(x)$... and $P(x)$

Let $m = n + 2k$ be number of points. For all points $i \in \{1, 2, \dots, m\}$,

$$Q(i) = P(i)E(i) \equiv r_i E(i) \pmod{p}$$

Gives $n + 2k$ linear equations:

From $Q(x)$		From $r_i E(x)$
$a_{n+k-1} + \dots a_0$	$\equiv$	$r_1(1 + b_{k-1} + \dots + b_0) \pmod{p}$
$a_{n+k-1}(2)^{n+k-1} + \dots a_0$	$\equiv$	$r_2((2)^k + b_{k-1}(2)^{k-1} + \dots + b_0) \pmod{p}$
		$\vdots$
$a_{n+k-1}(m)^{n+k-1} + \dots a_0$	$\equiv$	$r_m((m)^k + b_{k-1}(m)^{k-1} + \dots + b_0) \pmod{p}$

... and $n + 2k$ unknown coefficients of $Q(x)$ and $E(x)$!

Solve for coefficients of $Q(x)$ and $E(x)$.

$$\text{Find } P(x) = Q(x)/E(x).$$

How cool is that?!?!?

Example

Receiver gets: $r_1 = 3, r_2 = 1, r_3 = 6, r_4 = 0, r_5 = 3$

$$Q(x) = E(x)P(x) = a_3x^3 + a_2x^2 + a_1x + a_0$$

$$E(x) = x + b_0$$

$$Q(i) \equiv r_i E(i) \pmod{7}$$

$$a_3 + a_2 + a_1 + a_0 \equiv 3(1 + b_0) \pmod{7}$$

$$a_3 + 4a_2 + 2a_1 + a_0 \equiv 1(2 + b_0) \pmod{7}$$

$$6a_3 + 2a_2 + 3a_1 + a_0 \equiv 6(3 + b_0) \pmod{7}$$

$$a_3 + 2a_2 + 4a_1 + a_0 \equiv 0(4 + b_0) \pmod{7}$$

$$6a_3 + 4a_2 + 5a_1 + a_0 \equiv 3(5 + b_0) \pmod{7}$$

$a_3 = 1, a_2 = 6, a_1 = 6, a_0 = 5$, and $b_0 = -2$

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

$E(x) = x - 2$ ← Tells us error is at $i = 2$ How cool is that?!?!?!

Example: Finishing Up

$$Q(x) = x^3 + 6x^2 + 6x + 5 \quad \text{and} \quad E(x) = x - 2$$

$$\begin{array}{r} x^2 + x + 1 \\ \hline x - 2) x^3 + 6x^2 + 6x + 5 \\ x^3 - 2x^2 \\ \hline 1x^2 + 6x + 5 \\ 1x^2 - 2x \\ \hline x + 5 \\ x - 2 \\ \hline 0 \end{array}$$

$$P(x) = x^2 + x + 1 \pmod{7} \implies \text{Message is } P(1) = 3, P(2) = 0, P(3) = 6$$

Error Correction: Berlekamp-Welsh

This efficient decoding algorithm is the Berlekamp-Welch algorithm

After inventors Edwyn Berlekamp and Lloyd Welch

Berkeley Connection: Berlekamp was professor at Berkeley 1971–2002

Review...

Message: $m_1, m_2, \dots, m_n$

Sender:

Make degree $n - 1$ polynomial $P(x)$ where $P(i) = m_i$

Send $n + 2k$ values: $P(1), \dots, P(n + 2k)$

Receiver:

Receive $r_1, r_2, \dots, r_{n+2k}$

Solve $n + 2k$ equations, $Q(i) = r_i E(i)$ to find $Q(x) = E(x)P(x)$

Compute $P(x) = Q(x)/E(x)$

Compute $P(1), \dots, P(n)$

About the Computed Solution

Is there one and only one $P(x)$ from the Berlekamp-Welsh algorithm?

Existence (is there one?): There is a $P(x)$ and $E(x)$ that satisfy equations.

Unique solution for $P(x)$

Uniqueness (and only one): Any solution $Q'(x)$ and $E'(x)$ have

$$\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x). \quad (1)$$

Proof: We claim (proof on next slide!)

$$Q'(x)E(x) = Q(x)E'(x) \text{ on } n+2k \text{ values of } x \quad (2)$$

Equation (2) implies (1). Not as easy as it seems – **subtle issue** to handle:

Divide by $E(x)E'(x)$? For some x , $E(x)$ and/or $E'(x)$ might be zero!

$Q'(x)E(x)$ and $Q(x)E'(x)$ are degree $n+2k-1$ and agree on $n+2k$ points

$E(x)$ and $E'(x)$ have at most k roots each (recall: roots are error locations)

So n places where neither is zero: can cross divide at n points.

$$\implies \frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} \text{ equal on } n \text{ points} \quad (\text{Look Ma! No division by zero!})$$

Both degree $\leq n-1 \implies$ Same polynomial!



What About That Claim?

Claim: $Q'(x)E(x) = Q(x)E'(x)$ on $n+2k$ values of x .

Proof: Construction implies that for $i \in \{1, 2, \dots, n+2k\}$,

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

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

If $E(i) = 0$, then $Q(i) = 0$ and if $E'(i) = 0$, then $Q'(i) = 0$.

$\implies Q(i)E'(i) = Q'(i)E(i)$ holds when either $E(i)$ or $E'(i)$ is zero.

When $E'(i)$ and $E(i)$ are not zero (*don't divide by zero!*)

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

Cross multiplying gives

$$Q'(i)E(i) = Q(i)E'(i) = r_i,$$

for these points.

So holds when $E(i)$ is zero, $E'(i)$ is zero, or neither is zero.



Conclusion – and What Did We *Not* Say?

Proved: With $\leq k$ errors, $P(x) = \frac{Q(x)}{E(x)}$ is the unique solution, so reconstructs the message.

Yay!

Different: Are $Q(x)$ and $E(x)$ uniquely determined? Not necessarily....

Intuition: What if there are less than k errors?

$n = 4$, $k = 2$, transmit $n + 2k = 8$ packets, but an error only at packet 2

$E(x)$ is degree *exactly* $k = 2$ (degree 2 coefficient is 1)

Solving...

$E(x) = (x - 2)(x - 1)$ works (*If excess correct values, can blot them out!*)

$E(x) = (x - 2)(x - 2)$ works

$E(x) = (x - 2)(x - 3)$ works

...

Multiple valid solutions. Technically: rank of matrix is $< n + 2k$

Mentioned in the “Exercise” in the note.

Concept Check 1

Context:

You want to send a message of length 4

You construct $P(x)$ and send $P(1), P(2), \dots, P(8)$

Receiver gets $r_1, r_2, \dots, r_8$

Packets 1 and 4 are corrupted

Which of the following is not true?

- (A) $r_1 \neq P(1)$
- (B) The degree of $P(x)E(x)$ is 5
- (C) The degree of $E(x)$ is 2
- (D) The number of coefficients of $P(x)$ is 4
- (E) The number of coefficients of $Q(x)$ is 5

Concept Check 2

Context:

You want to send a message of length 4

You construct $P(x)$ and send $P(1), P(2), \dots, P(8)$

Receiver gets $r_1, r_2, \dots, r_8$

Packets 1 and 4 are corrupted

Which of the following are true?

- (A) $E(x) = (x - 1)(x - 4)$
- (B) The number of coefficients in $E(x)$ is 2
- (C) The number of unknown coefficients in $E(x)$ is 2
- (D) $E(x) = (x - 1)(x - 2)$
- (E) $r_4 \neq P(4)$

Summary

Erasure codes: Communicate n packets with k erasures.

How many packets to send? $n + k$

How to encode? With polynomial $P(x)$.

... of degree? $n - 1$

Recover? Reconstruct $P(x)$ with any n points!

Error Correcting Codes (ECC): Communicate n packets with k errors.

How many packets to send? $n + 2k$

How to encode? With polynomial $P(x)$.

... of degree? $n - 1$

Recover?

Reconstruct error polynomial, $E(x)$, and $P(x)$! **Nonlinear equations.**

Reconstruct $E(x)$ and $Q(x) = E(x)P(x)$. Linear Equations.

Polynomial division! $P(x) = Q(x)/E(x)$!

Reed-Solomon codes. Welsh-Berlekamp Decoding. Optimality. Perfection!