CS250/EE387 - LECTURE 5 - ALGORITHMS for REED-SOLOMON CODES!

AGENDA

0. Finishing up dual RS codes
1. Berlekamp-Welch
2. Berlekamp-Massey [sketch]

GASTROPOD FACT:

Most land slugs have two pairs of tentacles. The upper pair senses light, and the lower pair senses smell.

0. Recall the definition of RS codes:

DEF. (REED-SOLOMON CODES)

Let $n \geq k$, $q \geq n$. The REED-SOLOMON CODE of dimension k over $\mathbb{F}_q$, with evaluation points $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)$, is

$$RS_q(\vec{\alpha}, n, k) = \{ (f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n)) : f \in \mathbb{F}_q[X],\ \deg(f) \leq k-1 \}$$

Sometimes I'll just write $RS(n,k)$.

Last time, we saw that they meet the singleton bound.

[Need to finish up RS duality – see **LECTURE 4** notes].

HISTORIC ASIDE. RS codes were invented by Reed + Solomon in 1960. At the time, they didn't have any fast decoding algs, so they were sort of neat but not that useful. But in the late 1960's, Peterson, Berlekamp-Massey developed an $O(n^2)$-time alg, which can be made to run in time $O(n\log(n))$ with FFT tricks. Then RS codes started to be used all over the place! CDs, satellites, QR codes,... In 1986, Welch + Berlekamp came up w/ another decoding alg – it's a bit slower but it is really pretty, so we'll start with that.

① Welch-Berlekamp Algorithm

Problem (Decoding $RS_q(\vec{\alpha}, n, k)$ from $e \le \lfloor \frac{n-k}{2} \rfloor$ errors)

Given $w = (w_1, \dots, w_n) \in \mathbb{F}_q^n$, find a polynomial $f \in \mathbb{F}_q[X]$ so that:

- $\deg(f) < k$
- $f(\alpha_i) \neq w_i$ for at most $e \le \lfloor \frac{n-k}{2} \rfloor$ values of i,

or else return $\perp$ if no such polynomial exists.

Idea: Consider the polynomial $E(X) = \prod_{i : w_i \neq f(\alpha_i)} (X - \alpha_i)$.

This is called the "error locator polynomial." (Notice that we don't know what it is...)

Then $\forall i$, $w_i \cdot E(\alpha_i) = \underbrace{f(\alpha_i) \cdot E(\alpha_i)}_{\text{Call this } Q(\alpha_i)}$

Algorithm (Berlekamp-Welch)

① Find:

- a monic degree e polynomial $E(X)$
- a deg $\le e + k - 1$ polynomial $Q(X)$

so that: $w_i \cdot E(\alpha_i) = Q(\alpha_i) \; \forall i$ $\quad (*)$

If it doesn't exist, RETURN $\perp$.

② Let $\tilde{f}(X) = Q(X)/E(X)$

If $\Delta(\tilde{f}, w) > e$:

  RETURN $\perp$

RETURN $\tilde{f}$

TWO QUESTIONS:

1. How do we find such polys?
2. Once we do, why is it correct to return Q/E? What if we didn't find the "correct" Q and E?

Let's answer QUESTION 2 first.

CLAIM. If there is a degree $\leq k-1$ poly f s.t. $\Delta(f, w) \leq e$, then there exists E and Q satisfying $(*)$.

proof. Let $E(X) = \left[\prod_{i : w_i \neq f(\alpha_i)} (X - \alpha_i) \right] \cdot X^{e - \Delta(f,w)}$

Let $Q(X) = E(X) \cdot f(X)$.

CLAIM. Suppose that (E_1, Q_1), (E_2, Q_2) BOTH satisfy the requirements in STEP 1. Then:

$$\frac{Q_1(X)}{E_1(X)} = \frac{Q_2(X)}{E_2(X)}$$

proof. Consider $R(X) = \underbrace{Q_1(X)}_{\deg \leq e+k-1} \underbrace{E_2(X)}_{\deg e} - Q_2(X) E_1(X)$

$\deg(R) \leq 2e + k - 1$, and $\forall i \in \{1, \dots, n\}$,

$$R(\alpha_i) = [w_i \cdot E_1(\alpha_i)] \cdot E_2(\alpha_i) - [w_i \cdot E_2(\alpha_i)] \cdot E_1(\alpha_i) = 0$$

This is where we need $e \leq \lfloor \frac{n-k}{2} \rfloor$

Hence R has at least n roots. Since $e < \frac{n-k+1}{2}$, $2e + k - 1 < n$.
So $R \equiv 0$ is the all-zero polynomial. (Low degree polynomials don't have too many roots!) ■

Together, these CLAIMS answer QUESTION 2.

Moving on to QUESTION 1. How do we find E, Q? POLYNOMIAL INTERPOLATION!

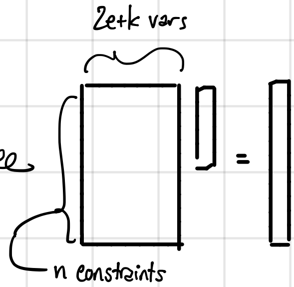
More precisely, we want:

$$\underbrace{w_i \cdot E(\alpha_i) = Q(\alpha_i) \text{ for } i = 1, \dots, n}_{n \text{ linear constraints.}} , \qquad \underbrace{\begin{array}{l} \deg(E) = e, \quad E \text{ monic} \\ \deg(Q) \leq e + k - 1. \end{array}}_{e + (e+k) = 2e+k \text{ variables,}}$$

which are the coefficients on these two polynomials.

We already know (from CLAIM 1) that a solution exists (assuming f does). So solve this system of eqs. to find it!

[Notice that $2e + k < (n-k+1) + k \leq n$, so the system looks like



But we don't actually care if the system is over or under-determined, now that we know that a solution exists and that any solution will do.]

RUNNING TIME of BERLEKAMP-WELCH:

- Step (1) takes time $O(n^3)$ for polynomial division
- Step (2) takes time $O(n^3)$ for Gaussian Elimination

$\Rightarrow O(n^3)$ total.

These notes adapted from Michael O'Sullivan's 2004 Lecture Notes from Math 696 at SDSU. [Available by Googling for them].

2 BERLEKAMP-MASSEY (sketch)

Again we solve this PROBLEM

PROBLEM (DECODING $RS_q(\vec{\alpha}, n, k)$ from $e \le \lfloor \frac{n-k}{2} \rfloor$ ERRORS)

Given $w = (w_1, \ldots, w_n) \in \mathbb{F}_q^n$, find a polynomial $f \in \mathbb{F}_q[X]$ so that:
- $\deg(f) < k$
- $f(\alpha_i) \ne w_i$ for at most $e \le \lfloor \frac{n-k}{2} \rfloor$ values of i,

or else return $\perp$ if no such polynomial exists.

The Berlekamp-Massey algorithm is more efficient than the Berlekamp-Welch alg, especially when the #errors is small. Also, it turns out to be really nice to implement in hardware, although we won't go into that.

Let H be the parity-check matrix for our RS code. I'm actually going to cheat a bit and add a row of ones on top, so that $H = G^T$ for some RS generator matrix G, since it makes the exposition a bit nicer. Everything in sight is a generalized RS code, so it doesn't matter too much.

So let

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \gamma & \gamma^2 & \gamma^3 & \cdots & \gamma^{n-1} \\ 1 & \gamma^2 & \gamma^4 & \gamma^6 & \cdots & \gamma^{2(n-1)} \\ \vdots & \vdots & & & & \\ 1 & \gamma^{n-k-1} & & \cdots & & \gamma^{(n-k-1)(n-1)} \end{bmatrix}$$

We will do SYNDROME DECODING (like for Hamming codes). That is, suppose $w = c + e$, and we can compute

$$H \cdot w = H \cdot c + H \cdot e = H \cdot e, \quad \text{since } H \cdot c = 0 \ \forall c \in \mathcal{C}.$$

Our goal will be to use $H \cdot e$ (the "SYNDROME") to recover $E(X)$, the error locator polynomial.

$$E(X) = \prod_{i : e_i \ne 0} (X - \gamma^i)$$

here I'm specializing to this particular order on eval pts b/c we picked H as above.

We don't have direct access to e, but we do have access to $H \cdot e$.

Consider, for some vector $(f_0, f_1, \ldots, f_{n-k-1})$,

$$\begin{bmatrix} f_0 & f_1 & \cdots & f_{n-k-1} \end{bmatrix} \begin{bmatrix} \\ & H & \\ \\ \end{bmatrix} \begin{bmatrix} \\ e \\ \\ \end{bmatrix}$$

This we can compute, since we know $H \cdot e$.
However, if we remember that $H = G^T$, this is also equal to

$$\underbrace{\begin{bmatrix} f(1) & f(\gamma) & f(\gamma^2) & \cdots & f(\gamma^{n-1}) \end{bmatrix} \begin{bmatrix} \\ e \\ \\ \end{bmatrix} =: \langle f, e \rangle \text{ or } \langle f(X), e \rangle}_{\text{INTRODUCING NOTATION.}}$$

where $f(X) = \sum_{i=0}^{n-k-1} f_i \cdot X^i$.

So, we can actually compute $\langle f, e \rangle$ for any f w/ $\deg(f) < n-k$.
Our goal is to use this power to recover e.

Actually, we are going to recover $E(X)$, then factor it to learn e.

OBSERVATION. $\boxed{\langle e, X^r \cdot E(X) \rangle = 0 \quad \forall r.}$

proof: $\langle e, X^r \cdot E(X) \rangle = \sum_{i=0}^{n-1} e_i \cdot \gamma^{i \cdot r} \cdot E(\gamma^i)$

one of these two things is zero.

OUR PLAN: Let's find some poly f s.t. $\deg(f) \leq t$ and $\langle e, X^r \cdot f(X) \rangle = 0$ for $r = 0, \ldots, t-1$.

It's not immediately clear that this is a good plan...

... but in fact it is a good plan:

PROP. Suppose that $wt(e) = t$, and that $\langle e, X^r \cdot f(X) \rangle = 0$ for $r = 0, \dots, t-1$. Then $E(X) \mid f(X)$.

In particular, if $\deg(f) \leq t$, $E(X) = \alpha \cdot f(X)$ for some $\alpha \in \mathbb{F}^*$.

Proof. If $\langle e, X^r \cdot f(X) \rangle = 0 \;\; \forall\, r = 0, \dots, t-1$, then by linearity,

$$\langle e, g(X) \cdot f(X) \rangle = 0 \quad \forall\; g \in \mathbb{F}_q[X] \text{ with } \deg(g) \leq t-1.$$

For any k s.t. $e_k = 1$, let $g_k(X) = \dfrac{E(X)}{X - \gamma^k} = \prod_{\substack{i : e_i \neq 0 \\ i \neq k}} (X - \gamma^i)$.

Then $\deg(g) \leq t-1$, hence

$$0 = \langle e, g(X) \cdot f(X) \rangle = \sum_{i=0}^{n-1} e_i \cdot g(\gamma^i) \cdot f(\gamma^i) = e_k \cdot g(\gamma^k) \cdot f(\gamma^k)$$

not zero

also not zero

better be zero!!

Hence, $f(\gamma^k) = 0$. So $(X - \gamma^k) \mid f(X) \;\; \forall\, k$ s.t. $e_k = 1$,
So $E(X) \mid f(X)$.

OK, so our plan is a good one. Let's try to find f so that:

- $\deg(f) \leq t$
- $\langle e, X^r \cdot f(X) \rangle = 0 \quad \forall\; r = 0, \dots, t-1.$

To this end, define:

$\mathrm{span}(f)$ = the smallest r s.t. $\langle e, X^r \cdot f(X) \rangle \neq 0$

$\mathrm{disc}(f) = \langle e, X^{\mathrm{span}(f)} \cdot f(X) \rangle$ is that nonzero value.

[This page skipped in class]

USEFUL LEMMA: If $\deg(g) \leq \operatorname{span}(f)$, then $\deg(g) + \operatorname{span}(g) \leq \deg(f) + \operatorname{span}(f)$.

Proof. First, suppose that $\deg(g) = \operatorname{span}(f)$, say $g(X) = \underset{\uparrow \text{nonzero}}{\alpha} \cdot X^{\operatorname{span}(f)} + \text{STUFF}$.

Then
$$\langle e, g(X)\cdot f(X)\rangle = \underbrace{\underbrace{\langle e, \underbrace{(g(X) - \alpha X^{\operatorname{sp}(f)})}_{\text{degree } < \operatorname{span}(f)} \cdot f(X)\rangle}_{\text{ZERO}} + \underbrace{\langle e, \alpha \cdot X^{\operatorname{sp}(f)} \cdot f(X)\rangle}_{\text{NONZERO by def of span}(f)}}_{\text{So this whole thing is NONZERO.}}$$

In particular, ONE of the terms that shows up in f, say X^c for $c \leq \deg(f)$, has $\langle e, g(X)\cdot X^c\rangle \neq 0$, hence $\operatorname{span}(g) \leq c \leq \deg(f)$.

Then $\deg(g) + \operatorname{span}(g) \leq \operatorname{span}(f) + \deg(f)$.

Next, if $\deg(g) < \operatorname{span}(f)$, apply the above to $X^{\operatorname{span}(f) - \deg(g)} \cdot g(X)$.

COR. If $\operatorname{span}(f) \geq t$ then $\operatorname{span}(f) = \infty$.

proof. Say $\operatorname{span}(f) \geq t$ but is finite. Then $\deg(E) = t \leq \operatorname{span}(f)$, so by the USEFUL LEMMA,

$$\deg(E) + \underbrace{\operatorname{span}(E)}_{\infty} \leq \underbrace{\deg(f) + \operatorname{span}(f)}_{\text{finite.}}$$

CONTRADICTION!

Again, our goal is to come up w/ some function w/ large span. The following lemma will tell us how to get this.

LEMMA Suppose $\text{span}(f) = r$, $\text{disc}(f) = \mu$ ← RECALL, this means that $\langle e, X^r f(X) \rangle = \mu$, and $\langle e, X^i f(X) \rangle = 0 \ \forall i < r$.

Suppose $\text{span}(g) = c$, $\text{disc}(g) = \nu$

AND say that $c \leq r$.

Then $h(X) = f(X) - \left(\frac{\mu}{\nu}\right) \cdot X^{c-r} \cdot g(X)$ has

$\text{span}(h) > \text{span}(f)$.

The point of this lemma is that, given f and g with reasonably close spans, we can combine them to get h w/ a strictly bigger span and degree not too much larger.

Pf. Just consider

$$\left\langle e, X^i \cdot \left[f(X) - \left(\frac{\mu}{\nu}\right) X^{c-r} g(X) \right] \right\rangle$$

$$= \langle e, X^i f(X) \rangle - \left(\frac{\mu}{\nu}\right) \langle e, X^{c-r+i} \cdot g(X) \rangle$$

If $i < r$, then both terms are 0 since $sp(f) = r$, $sp(g) = c$.
If $i = r$, then we have $\mu - \left(\frac{\mu}{\nu}\right) \cdot \nu = 0$.

Hence $sp(h) > r$.

ALGORITHM (BERLEKAMP-MASSEY):

Initialize $f \leftarrow 1$, $g \leftarrow 0$

for $m = 0, \dots, 2t-1$:

$c \leftarrow \deg(f) - 1$

$r \leftarrow m - c - 1$ $(= m - \deg(f))$

$\mu \leftarrow \langle e, X^r \cdot f(X) \rangle$

RECALL, we can compute this as long as $\deg(X^r \cdot f(X)) \le n-k-1$

[This will impose the constraint $2t-1 \le n-k-1$]

if $\mu = 0$ OR $r \le c$:

$f'(X) \leftarrow f(X) - \mu \cdot X^{c-r} \cdot g(X)$

$g'(X) \leftarrow g(X)$

We will maintain inductive hyps that imply that in this case, $\mathrm{span}(f) = r$, $\mathrm{span}(g) = c$, $\mathrm{disc}(g) = 1$, so this is just the update from the LEMMA.

else:

$f'(X) \leftarrow X^{r-c} \cdot f(X) - \mu \cdot g(X)$

$g'(X) \leftarrow \frac{1}{\mu} \cdot f(X)$

In this case we'll have $\mathrm{span}(f) = r$, $\mathrm{span}(g) = c$, and $\mathrm{disc}(g) = 1$, so the LEMMA's update would be

$h \leftarrow g(X) - \frac{1}{\mu} \cdot X^{r-c} f(X)$,

and this is just $-\mu$ times that.

[It makes sure we keep $\mathrm{disc}(g) = 1$].

$f, g \leftarrow f', g'$

RETURN $f(X)$

CLAIM. This algorithm maintains:

After iteration m:

- f is monic and $\deg(f) + \mathrm{span}(f) > m$
- EITHER $g = 0$

  OR:
  - $\mathrm{span}(g) = \deg(f) - 1$
  - $\mathrm{span}(g) + \deg(g) \le m$
  - $\mathrm{disc}(g) = 1$.

[This proof skipped in class]

Initialize $f \leftarrow 1$, $g \leftarrow 0$

for $m = 0, \dots, 2t-1$:

$c \leftarrow \deg(f) - 1$

$r \leftarrow m - c - 1 = m - \deg(f)$

$\mu \leftarrow \langle e, X^r \cdot f(X) \rangle$

if $\mu = 0$ OR $r \leq c$:

$f'(X) \leftarrow f(X) - \mu \cdot X^{c-r} \cdot g(X)$

$g'(X) \leftarrow g(X)$

else:

$f'(X) \leftarrow X^{r-c} \cdot f(X) - \mu \cdot g(X)$

$g'(X) \leftarrow \frac{1}{\mu} \cdot f(X)$

$f, g \leftarrow f', g'$

RETURN $f(X)$

Alg. repeated for the reader's (and writer's...) convenience.

Proof. The base case (after $m = -1$) is easy.

Let $m \geq -1$ and assume by induction that

(1) f is monic and $\deg(f) + \mathrm{span}(f) > m$

(2) EITHER $g = 0$ OR:

- $\mathrm{span}(g) = \deg(f) - 1$
- $\mathrm{span}(g) + \deg(g) \leq m$
- $\mathrm{disc}(g) = 1$.

CASE 1. $\mu = 0$. Then f and g are unchanged.

So the stuff (2) about g is good.

Further, since $0 = \mu = \langle e, X^r f(X) \rangle = \langle e, X^{m - \deg(f)} \cdot f(X) \rangle$,

we have $\mathrm{span}(f) > m - \deg(f)$, hence $\mathrm{span}(f) + \deg(f) > m$, so (1) holds.

CASE 2. $\mu \neq 0$.

CASE 2A. $r \leq c$.

$\mathrm{span}(g) \overset{\text{induction}}{=} \deg(f) - 1 = c$ by our choice of c.

Hence $\mathrm{span}(X^{c-r} \cdot g(X)) = r$.

Then $f' \leftarrow f(X) - \mu \cdot X^{c-r} \cdot g(X)$.

$\deg(f) + \mathrm{span}(f) \overset{\text{induction}}{\geq} m \overset{\text{choice of } r}{=} \deg(f) + r \Rightarrow \mathrm{span}(f) \geq r$.

And since $\mu \neq 0$, $\mathrm{span}(f) = r$.

So both f, g have span $= r$, $\mathrm{disc}(f) = \mu$, $\mathrm{disc}(g) = 1$, so this update is precisely the one from the LEMMA and $sp(f') > r$.

Moreover, $\deg(f') = \deg(f)$, hence $\deg(f') + sp(f') > \deg(f) + sp(f) > m$

$\Rightarrow \deg(f') + sp(f') > m+1$

To see this, notice that $\deg(g) = (sp(g) + \deg(g)) - sp(g) \leq m - \mathrm{span}(g) = m - c$

So $\deg(X^{c-r} g(X)) \leq (m-c) + (c-r) = m - r < \underbrace{\deg(f)}$.

b/c $\mathrm{span}(f) = r$, and by induction $sp(f) + \deg(f) > m$.

Thus, the update "$-\mu \cdot X^{c-r} g(X)$" affects neither the degree, nor the monicness of f

This arg says also that f' is monic, so (1) holds for $m+1$.

(2) holds since in this case we did not update g.

CASE 2B. $c < r$ is similar. [FUN EXERCISE].

COR. Suppose $wt(e) = t$.
If $m \geq 2t - 1$, then after iteration m, $f(X) = E(X)$.

proof. First notice that $\deg(f(X)) \leq t$.

Indeed, we've been maintaining $span(g) = \deg(f) - 1$, so if $\deg(f) > t$ then $sp(g) \geq t$.
By the USEFUL LEMMA (or rather, its COR), we have $span(g) = \infty$.
But we were also maintaining $span(g) + \deg(g) \leq m$, so that's a ↯.

Now,
$$\deg(f) + span(f) > m$$
$$\Rightarrow \quad span(f) > m - \deg(f)$$
$$\geq (2t-1) - t$$
$$= t - 1$$
So $span(f) \geq t$.

But this is what we wanted:

By earlier LEMMA.

$$sp(f) \geq t \Rightarrow E(X) \mid f(X)$$
$$\deg(f) \leq t \Rightarrow E(X) = \alpha \cdot f(X) \quad \text{for some } \alpha \in \mathbb{F}^*$$
$$f \text{ monic} \Rightarrow E(X) = f(X).$$

Finally, recall that $d = n - k + 1$, and that the algorithm stops working (we stop being able to query $\langle e, X^i \cdot f(X) \rangle = \mu$) when $m \geq n - k$, so we need $2t - 1 \leq n - k - 1$

aka $\quad t \leq \frac{n-k}{2} = \frac{d-1}{2}$ which is where the algorithm should stop working.

HOWEVER! Notice that if t happens to be smaller, we can actually stop earlier, with only $O(t)$ rounds. The polys we are working with all have $\deg \leq m = O(t)$, and so we can do everything in $poly(t)$ computations over $\mathbb{F}_q$. That's sublinear time!!

[See "Syndrome Encoding and Decoding of BCH Codes in Sublinear Time" by Dodis, Ostrovsky, Reyzin, Smith for details about making this real fast.]

All this just finds $E(X)$. We still need to find the roots of $E(X)$, and then figure out how to fix the errors.

- If you get fancy, you can factor $E(X)$ in time $O(t^{1.8ish} \cdot \log(n))$
  ["Subquadratic-time factoring of polynomials over finite fields" Kaltofen + Shoup 1995]

- To actually recover the message, we can't hope for sublinear time (since the message has length $k = Rn$), but we can now do that in time $O(n\log(n))$ via linear algebra. [The $n\log(n)$ is b/c Vandermonde matrices admit a nice FFT-like alg.]

That finishes the Berlekamp-Massey algorithm.

This algorithm can actually be implemented nicely in hardware [the update step can be done with a shift register] and so this is the alg. that's often used in practice for RS codes. (Or, optimized versions of this).

The Berlekamp-Welch alg is certainly easier to understand, though!

QUESTIONS TO PONDER

1. Fill in the details for the Berlekamp-Massey alg.
   [there is one FUN EXERCISE in the notes and I anticipate we skipped some proofs in class]

2. Can you think of any other algs for RS codes?

3. How would you adapt RS codes / these algorithms to come up with BINARY codes?