April 16, 2019

REMINDERS

(1) Mini project video due in 2 weeks (11:59pm, April 30)
(2) HW 3 has been graded
(3) 3rd proof reading schedule is now up.

RS Unique decoding

$\alpha_1, \ldots, \alpha_n$ distinct

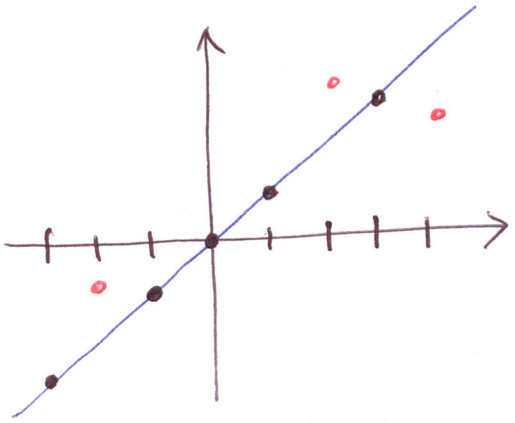
PROBLEM: Input: $(y_1, \alpha_1), (y_2, \alpha_2), \ldots, (y_n, \alpha_n) \in \mathbb{F}_q \times \mathbb{F}_q$

Output: (Unique) $P(X)$ of deg $< k$ s.t for at least $t = n - e$ locations $i \in [n]$ $y_i = P(\alpha_i)$ where

$$e < \frac{n-k+1}{2} \quad \text{or} \quad t > \frac{n+k-1}{2}$$

we're looking for "lines"

For example: $n = 8, k = 2, e = 3$



Welch-Berlekamp algo outline:

1. Assume you know $P(X)$
2. Come up w/ some poly identity that $P(X)$ satisfies
3. Generalize identity & "forget" $P(X)$,
4. "Solve" general identity to get $P(X)$

Error-locator poly

$E(X)$ s.t $\forall i$ s.t $P(\alpha_i) \neq y_i \Rightarrow E(\alpha_i) = 0$

$\Rightarrow$ E.g. $E(X) = \prod_{i : P(\alpha_i) \neq y_i} (X - \alpha_i)$

$\Rightarrow$ CLAIM: $\forall i \in [n]$ $P(\alpha_i) E(\alpha_i) = y_i E(\alpha_i)$

n equations. BUT quadratic eqns!

vars: k + e $\Rightarrow$ n eqns in $k+e \ll n$ vars

TODAY [Proof reader: Alex]

1. Finish WB algo
2. Start with list decoding algos for RS

$\forall\, i \in [n] \quad y_i E(\alpha_i) = P(\alpha_i) E(\alpha_i)$

Linearization: $N(X) = P(X) E(X)$ $\quad \deg(N) \le k-1+e = k+e-1$

1) $\forall\, i \in [n] \hookrightarrow N(\alpha_i) = y_i E(\alpha_i)$

→ n linear eqns

→ #vars = $k+e+e = k+2e$

we'll OK if #vars ≤ #eqns.

$k+2e \le n$

or $e \le \frac{n-k}{2}$

2) If we compute $N(X)$ & $E(X)$

hopefully $P(X) = \frac{N(X)}{E(X)}$

Welch-Berlekamp Algo

i/p: $1 \le k \le n$, $0 < e < \frac{n-k+1}{2}$, n pairs (α_i, y_i)

o/p: $P(X)$ of deg $\le k-1$ or FAIL

(1) Compute a monic $E(X)$ of deg $= e$, $N(X)$ of deg $\le k+e-1$ s.t. $\forall\, i \in [n] \quad N(\alpha_i) = y_i E(\alpha_i)$ } Q2 on HW0 $O(n^3)$

(2) If no such $N(X)$ or $E(X)$ $\exists$, output FAIL

(3) If $E(X)$ does not divide $N(X)$ } "long division" $O(n^3)$

(4) $P(X) \leftarrow \frac{N(X)}{E(X)}$

(5) If $\deg(P) \ge k$ or $\Delta\left(\bar{y}, (P(\alpha_i))_{i=1}^{n}\right) > e$ o/p FAIL

(6) Output $P(X)$

If WB outputs $P(X)$ ⇒ WB is correct.

THM: If $P(X)$ is close enough to $\bar{y}$ ⇒ WB outputs $P(X)$

Claim 1: $\exists\ N^*(X), E^*(X)$ satisfying step (1) s.t. $P(X) = \frac{N^*(X)}{E^*(X)}$

($\deg \le e+k-1$; E^* monic deg e)

$\forall i\ N(\alpha_i) = y_i E(\alpha_i)$

Claim 2: $\exists\ (N_1(X), E_1(X)) \neq (N_2(X), E_2(X))$ that both satisfy step 1 $\Rightarrow \frac{N_1(X)}{E_1(X)} = \frac{N_2(X)}{E_2(X)}$

Claims 1+2 $\Rightarrow$ Thm

$e\text{-}A(\bar{y}, (P(\alpha_i))_{i=1}^n)$

Pf of Claim 1: $E^*(X) = X^{\,\cdot} \cdot \prod_{i:\, P(\alpha_i) \neq y_i} (X - \alpha_i)$ ← monic + deg $= e$

$N^*(X) = E^*(X) \cdot P(X)$ ← deg $\le e+k-1$

Also $N^*(\alpha_i) = y_i \cdot E(\alpha_i)$ $\forall i \in [n]$

Pf of Claim 2: Use degree mantra. $\forall i$ $N_1(\alpha_i) = y_i E_1(\alpha_i)$, $N_2(\alpha_i) = y_i E_2(\alpha_i)$

$$R(X) = N_1(X)E_2(X) - N_2(X)E_1(X)$$

$\forall i$, $R(\alpha_i) = N_1(\alpha_i)E_2(\alpha_i) - N_2(\alpha_i)E_1(\alpha_i)$

$= y_i E_1(\alpha_i)E_2(\alpha_i) - y_i E_2(\alpha_i)E_1(\alpha_i)$

$= 0$

$\Rightarrow R$ has n roots.

$\deg(R) \le \deg(N) + \deg(E) \le e+k-1+e = 2e+k-1 < n$ (by choice of e)

$\Rightarrow$ By deg mantra $R(X) = 0$

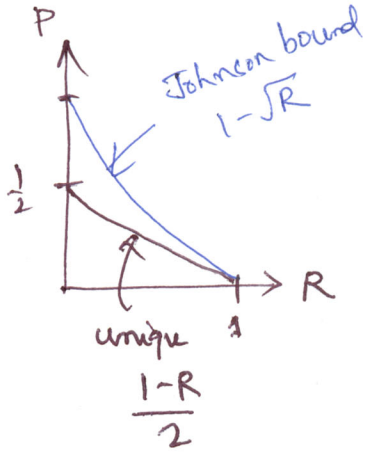
$\Rightarrow N_1(X)E_2(X) = N_2(X)E_1(X)$

$\Rightarrow \frac{N_1(X)}{E_1(X)} = \frac{N_2(X)}{E_2(X)}$ (since $E_1(X), E_2(X) \neq 0$ as $\deg(E_1) = \deg(E_2) = e > 0$)

Observed WB takes $O(n^3)$ ops over $\mathbb{F}_q$

Thm: For any $[n, k, d]_q$ RS code, correct up to $\le \frac{d-1}{2} = \frac{n-k}{2}$ error is $O(n^3)$ ops over $\mathbb{F}_q$.

List decoding for RS codes



Recall: Johnson bound $\Rightarrow$ RS is $(1-\sqrt{R}, O(n^2))$ - list decodable.

Q: Can we correct upto $1-\sqrt{R}$ frac of error in poly(n) time?

A: Yes.

Plan: Algo 1: $1-2\sqrt{R}$, Algo 2: $1-\sqrt{2R}$ ↑ Sudan '96, Algo 3: $1-\sqrt{R}$ ↑ Guruswami-Sudan '98

I/p: $(y_1, \alpha_1), \ldots, (y_n, \alpha_n) \in \mathbb{F}_q \times \mathbb{F}_q$

O/p: ALL poly $P(X)$ of deg $\leq k-1$ s.t $P(\alpha_i) = y_i$ for at least t location $i \in [n]$

↑ make this as ~~large~~ small ~~as~~ as possible

WB $\Rightarrow$ $t \geq \frac{n+k}{2}$

WB reformulation

Step 1: Find $P(X)$ & $N(X)$ s.t $\forall i \in [n]$
$P(X)$: ↑ deg $< k$; $N(X)$: ↑ deg $\leq e+k-1$; $N(\alpha_i) = y_i \cdot E(\alpha_i)$

Step 2: If $Y - P(X)$ divides $Q(X,Y) = Y E(X) - N(X)$
$\Rightarrow$ output $P(X)$

$\equiv$ $P(X) = \frac{N(X)}{E(X)}$

→ Step 1': Compute $Q(X,Y) = Y E(X) - N(X)$
s.t $\forall i \in [n]$, $Q(\alpha_i, y_i) = 0$

LD algo skeleton:

Step 1 (interpolation): Compute non-zero $Q(X,Y)$ s.t $\forall i \in [n]$ $Q(\alpha_i, y_i) = 0$

Step 2 (root finding): If $Y - P(X)$ divides $Q(X,Y)$ • add $P(X)$ to output list

WB: $Q(X,Y) = Y E(X) - N(X)$

Algo 1/2/3: Put less constraints on $Q(X,Y)$

Step 2: $\exists$ poly time algo to factorize $Q(X,Y)$

↳ Run this & retain all linear factor $Y - P(X)$

Def: $\deg_X(Q(X,Y))$ is the largest exponent of X in any monomial

$\deg_Y(Q(X,Y))$ ——————— Y ———

$\deg_X(X^2Y + Y^2X) = 2$

$\deg_Y(X^2Y + Y^2X) = 2$

Algo 1:

Do step 1 s.t $\deg_X(Q) \leq \ell$, $\deg_Y(Q) \leq \frac{n}{\ell}$

↳ $\forall i \in [n]$ $Q(x_i, y_i) = 0$

↳ n linear equations in coefficients of Q.

coeff of $Q = (\ell+1)\left(\frac{n}{\ell}+1\right) > n$