ALGERRA IN ALGORITHMIC CODING THEORY

http://madhu.seas.harvarel.edu

1. • INTRODUCE BASIC ALGORITHMIC PROBLEMS

Decoding, List-Decoding

• SINGLETON BOUND, REED-SOLOMON CODES,

DECODING RS CODES.

2. LIST-DECODING CAPACITY;

ACHIEVING CAPACITY WITH FOLDED RS CODES

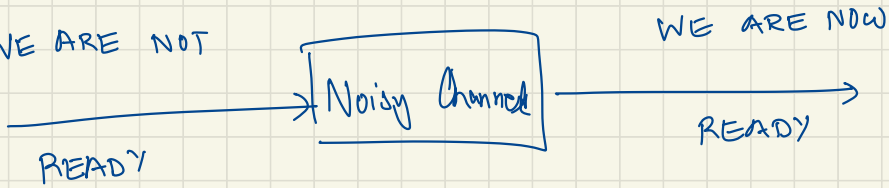
3. LOCAL DECODING PROBLEM

REED-MULLER CODES + LOCAL DECODING

4. MULTIPLICITY CODES

MATCHING VECTOR CODES.

Error-Correction Problem: Motivation ('40s)



- Digital Information: Meaning Changes Discontinuously as function of "distance".
- Need to Protect. What Protects?

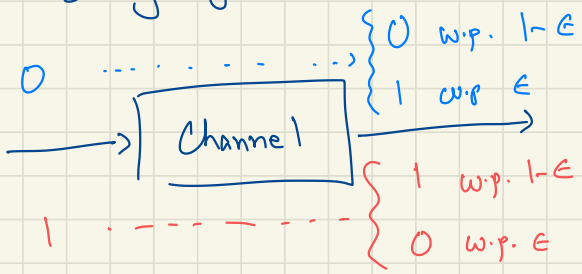
Shannon '48: Mathematical Theory of Communication

Hamming '50: Theory of Error-Detecting & Error-Correcting Codes.

Modelling the Problem

Shannon / Random: Channel: maps $\Sigma \to \Sigma$ independently on each use.

Eg "Binary Symmetric Channel" (BSC(ϵ))



Hamming / Adversarial: Channel: $\Sigma^n \to \Sigma^n$ adversarially subject to "distance $\leq \epsilon n$"

HAMMING DISTANCE $\Delta : \Sigma^n \times \Sigma^n \to \mathbb{Z}^{\geq 0}$

$$\Delta(x,y) \triangleq \#\{i \mid x_i \neq y_i\}$$

$x_1 \dots x_n$ $y_1 \dots y_n$

$$\delta(x,y) \triangleq \frac{\Delta(x,y)}{n}$$ "relativized Hamming distance"

$$\delta : \Sigma^n \times \Sigma^n \to [0,1]$$

Definitions:

Encoding: $E: \Sigma^k \to \Sigma^n$ ← injective (one-to-one)

Decoding: $D: \Sigma^n \to \Sigma^k$

Desire: $D(\text{Channel}(E(m))) = m$ (w.h.p. over channel.)

General Intuition: E is "good" if related code C is "good".

Code: $C = \text{Image}(E) = \{E(m) \mid m \in \Sigma^k\}$

$(C \subseteq \Sigma^n)$

Key Parameters:

Rate $(C/E) \triangleq \frac{k}{n} \triangleq \frac{\text{message length}}{\text{block length}}$

Distance (C/E): $\Delta(C) \triangleq \min_{x \neq y \in C} \{\Delta(x,y)\}$

$\delta(C) \triangleq \min_{x \neq y \in C} \{\delta(x,y)\}$

Alphabet = ?: Extremes $\Sigma = \{0,1\}$; $|\Sigma| = \text{poly}(n)$.

Algorithmic Problems

① Encode $E: \Sigma^k \to \Sigma^n$

Usually easy by choice of E ("We" choose E)

② Decode $D: \Sigma^n \to \Sigma^k$

- Guarantee expected: if $\Delta(E(m), x) \leq e$ then $D(x) = m$.
- Need to assume $e < \frac{\Delta(E)}{2}$. (why?)
- if we assume $e < \frac{\Delta(E)}{2}$ then m unique (why?)

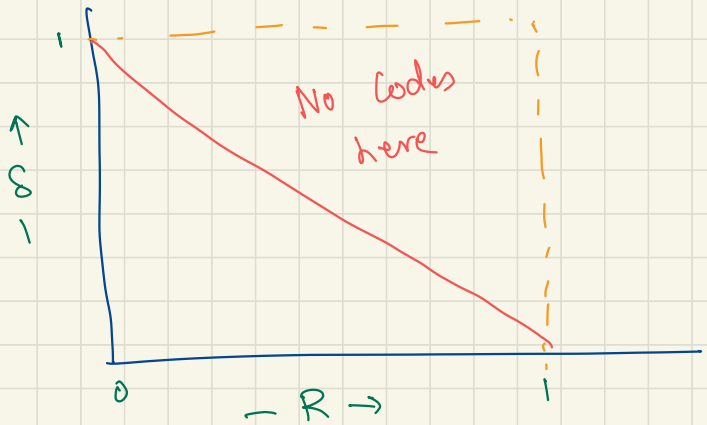
③ List-Decode $D: \Sigma^n \to (\Sigma^k)^L$

- Expected Guarantee: $\Delta(E(m), x) \leq e \Rightarrow m \in D(x)$.
- D computable in poly(n) time $\Rightarrow L \leq$ poly(n)
- Hope $e > \frac{\Delta(E)}{2}$! maybe even $e > \frac{n}{2}$!

Rate vs. Distance

$\forall\ C,\quad 0 \le R(C) \le 1,\quad 0 \le \delta(C) \le 1$

higher R, δ nicer.



Singleton Bound (Pigeonhole Principle):

- $E: \Sigma^k \to \Sigma^n \Rightarrow \quad k + \Delta(E) \le n+1$

$\Leftrightarrow \forall C \quad R(C) + \delta(C) \le 1 + \frac{1}{n+1}$

Proof: let $\pi: \Sigma^n \to \Sigma^{k-1}$ send $(x_1 .. x_n) \mapsto (x_1 .. x_{k-1})$

Then $\pi \circ E: \Sigma^k \to \Sigma^{k-1}$ not injective

$\Rightarrow \exists\ x \neq y \in \Sigma^k$ s.t. $\pi(E(x)) = \pi(E(y))$

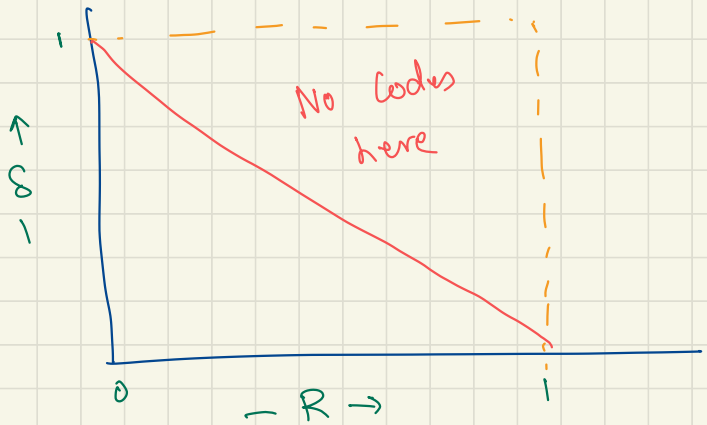
$\Rightarrow \exists\ E(x), E(y) \in C$ s.t. $\Delta(E(x), E(y)) \le n - (k-1)$

$\boxtimes$

Rate vs. Distance

$\forall\ C,\quad 0 \le R(C) \le 1,\quad 0 \le \delta(C) \le 1$

higher R, δ nicer.



Singleton Bound (Pigeonhole Principle):

- $E: \Sigma^k \to \Sigma^n \Rightarrow \quad k + \Delta(E) \le n+1$

$\Leftrightarrow \forall C \quad R(C) + \delta(C) \le 1 + \frac{1}{n+1}$

Proof: let $\pi: \Sigma^n \to \Sigma^{k-1}$ send $(x_1 .. x_n) \mapsto (x_1 .. x_{k-1})$

Then $\pi \circ E: \Sigma^k \to \Sigma^{k-1}$ not injective

$\Rightarrow \exists\ x \neq y \in \Sigma^k$ s.t. $\pi(E(x)) = \pi(E(y))$

$\Rightarrow \exists\ E(x), E(y) \in C$ s.t. $\Delta(E(x), E(y)) \le n - (k-1)$

$\boxtimes$

Reed-Solomon Codes:

$\Sigma = \mathbb{F}_q \triangleq$ finite field with q elements

Parameters: $k \leq n \leq q$

$\alpha_1 \dots \alpha_n$ distinct $\alpha_i \in \mathbb{F}_q$

Message: $m = (m_0, \dots m_{k-1}) \in \mathbb{F}_q^k$

Equated with $M(x) = \sum m_i x^i$

Encoding: $(M(\alpha_1), \dots M(\alpha_n)) \in \mathbb{F}_q^n$

Distance: deg $k-1$ non-zero poly has at most $k-1$ roots

$\Rightarrow \Delta(\text{Reed-Solomon}) \geq n-k+1$

↑ Matches Singleton!!

Warning: $q \geq n$

Smaller q?

① Existence: $\forall \epsilon < 1/2 \quad \exists R > 0 \quad \forall$ large n

$$\exists E : \{0,1\}^k \to \{0,1\}^n$$

s.t. $\frac{k}{n} \geq R$, $\delta(E) \geq \epsilon$

$$R = 1 - H(\epsilon) \; ; \quad H(\epsilon) = \epsilon \log_2 \frac{1}{\epsilon} + (1-\epsilon) \log_2 \frac{1}{1-\epsilon}$$

Proof: Pick E at random ....

② E, D as above computable in $\text{Time}(2^n)$

Proof: Pick $E(x) = T \cdot x$

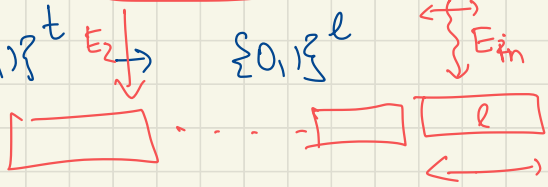
for random Toeplitz matrix.

③ Can combine "concatenate" Reed-Solomon + ② above to get polytime E, D. with Rate, $\delta > 0$.

Concatenation: View $\mathbb{F}_q \cong \{0,1\}^t$ $\quad t = \log_2 q$

- RS Code: $E_{RS} : \{\{0,1\}^t\}^k \to \{\{0,1\}^t\}^n$

- "Inner" Code: $E_{in} : \{0,1\}^t \to \{0,1\}^\ell$ (from ②)



- Combined "Concatenated" Code

$$E_{comb}(m_0, \ldots m_{k-1}) = E_{in}(y_1), E_{in}(y_2) \ldots E_{in}(y_n)$$

where $(y_1 \ldots y_n) = E_{RS}(m_0 \ldots m_{k-1})$

- Lemma: $\delta(E_{comb}) = \delta(E_{RS}) \cdot \delta(E_{in})$

$$R(E_{comb}) = R(E_{RS}) \cdot R(E_{in})$$

EXERCISE

Moral: Codes over $q = n^{O(1)}$ good enough for us.

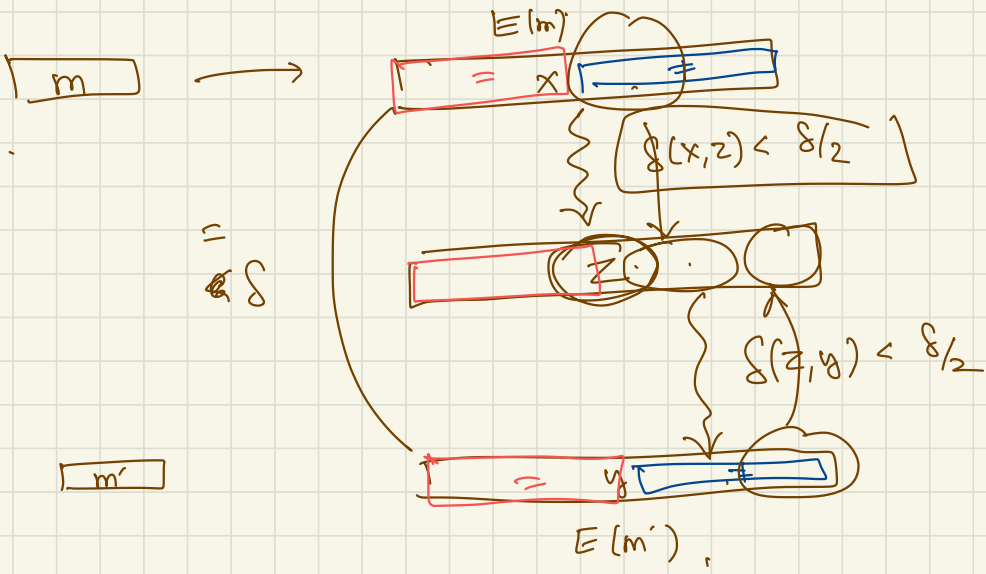
Distance vs. Error-Correction

Prop [Hamming]: Code of distance δ corrects $\delta/2$ fraction adversarial errors

UNIQUE DECODING

Proof by picture: (in class)

Key $\delta(x,y) \leq \delta(x,z) + \delta(z,y)$



But ... unique decoding not the right objective!

List-decoding

- In English: let decoder output small list of codewords.
  "Success" $\triangleq$ List includes transmitted codeword.

- Math: $E: \Sigma^k \to \Sigma^n$

  $D: \Sigma^n \to (\Sigma^k)^L$ ← set of size L.

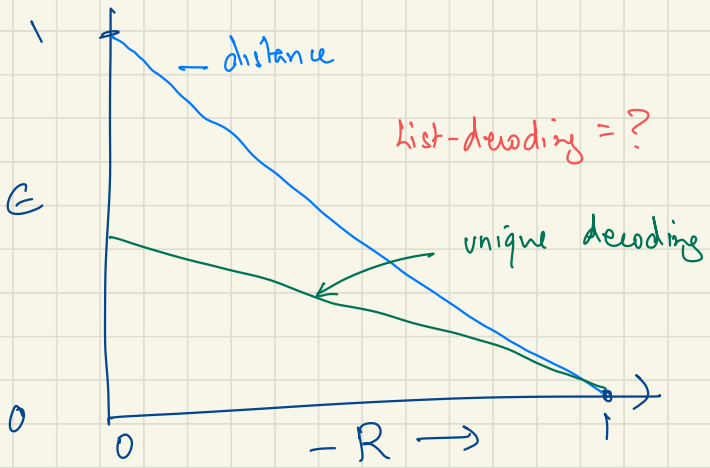
Code E is (ϵ, L)-list-decodable if $\exists D$

$\forall\ m, y$ s.t. $\delta(E(m), y) \leq \epsilon$

$m \in D(y)$.

Algorithmic: if E, D polytime computable.

Limits of Error-Correction



List-Decoding of Reed Solomon Codes

Given: $n, k, \mathbb{F}_q, \alpha_1 \dots \alpha_n, \epsilon$

$\beta_1 \dots \beta_n$

Output: All polynomials p s.t. $\deg(p) < k$

& $\left| \{ i \mid \beta_i = p(\alpha_i) \} \right| \geq (1-\epsilon) n$

$[\epsilon > \frac{1}{2}?]$

Algorithm: I. find $Q(x,y) \not\equiv 0$ s.t. $\deg_x Q < \sqrt{n}$

$\deg_n Q < \sqrt{n}$

$Q(x,y) = \sum c_{ij} x^i y^j$

$Q(\alpha_i, \beta_i) = 0 \quad \forall i$

II. Factor $Q(x,y)$ & report all p s.t.

$y - p(x) \mid Q(x,y)$

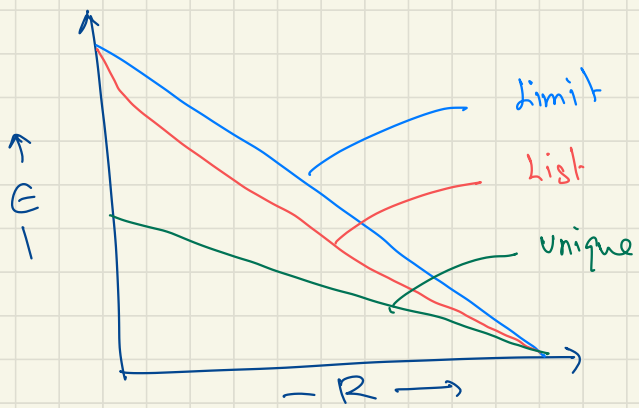
Lemma 1: Such Q always exists

Lemma 2: if $\left| \{ i \mid p(\alpha_i) = \beta_i \} \right| > 2k\sqrt{n}$

$\Rightarrow \quad y - p(x) \mid Q(x,y)$

Solves problem if $\epsilon < 1 - \frac{2k}{\sqrt{n}}$

- Can improve to $\epsilon < 1 - 2\sqrt{\frac{k}{n}}$ by better parameters
- Can improve to $\epsilon < 1 - \sqrt{\frac{2k}{n}}$ by more careful design of Q. [S. '97]
- Can improve to $\epsilon < 1 - \sqrt{\frac{k}{n}}$ with much more ideas. [Guruswami + S. '98]

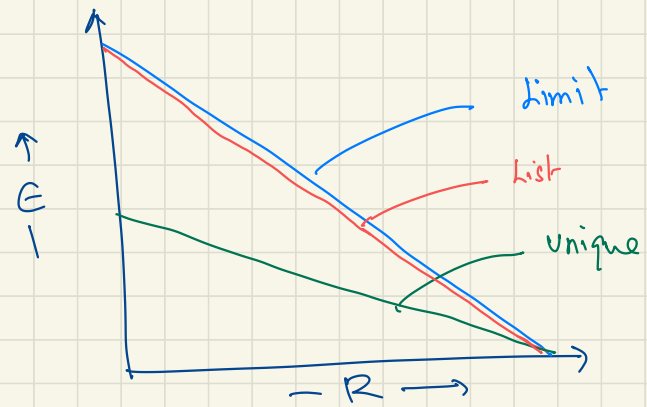


Next lecture

$\forall \epsilon > 0, \delta > 0 \quad \exists E \quad$ s.t.

E is $(\epsilon, \text{poly}(n))$ - list-decodable

& $\text{Rate}(E) \geq 1 - \epsilon - \delta$.



Algebra In Algorithmic Coding Theory - 2

Recall:

List-decoding

- In English: let decoder output small list of codewords.
  "Success" $\triangleq$ List includes transmitted codeword.

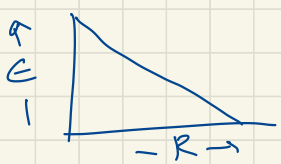
- Math: $E: \Sigma^k \to \Sigma^n$

  $D: \Sigma^n \to \left(\Sigma^k\right)^L$ ← set of size L.

Code E is (ϵ, L)-list-decodable if $\exists D$

$\forall\ m, y$ s.t. $\delta(E(m), y) \leq \epsilon$

$m \in D(y)$.



Algorithmic: if E, D polytime computable.

Today:

$E: \Sigma^k \to \Sigma^n$

$|\Sigma| = q = n^{\frac{1}{\delta^2}}$

$\forall\ \epsilon > 0, \delta > 0$ $\exists\ E$ s.t.

E is $(\epsilon, \text{poly}(n))$-list-decodable

& $\text{Rate}(E) \geq 1 - \epsilon - \delta$.

Aside: Brief History

1. Bleichenbacher, Kiayias, Yung: Crypto scheme + attack
2. Coppersmith, Sudan: Stronger attack
3. Parvaresh, Vardy: Novel Coding
4. Guruswami, Rudra: Key Optimization
5. Salil Vadhan: reminder of BKY
6. Guruswami-Wang: Simpler proof, more general construction

The codes: Folded Reed-Solomon Codes

Message $= P_m \in \mathbb{F}_q^k \approx \mathbb{F}_q^{<k}[X]$

↑ polys of deg $< k$.

Parameters: $\gamma \in \mathbb{F}_q$; $\alpha_1 \ldots \alpha_n \in \mathbb{F}_q$; Integer $m > 0$

$\{\alpha_i\}$'s all distinct

Encoding

$P(\alpha_1), P(\gamma \cdot \alpha_1), \ldots P(\gamma^{m-1}\alpha_1)$ . . . . . $P(\alpha_n) \ldots P(\gamma^{m-1}\alpha_n)$

$E : \mathbb{F}_q^k \to \left(\mathbb{F}_q^m\right)^n$ $\quad \Sigma = \mathbb{F}_q^m$

$\Sigma = \mathbb{F}_q^m$

$\mathbb{F}_q^k \to \Sigma^n$; $\Sigma^{\frac{k}{m}} \to \Sigma^n$

Lemma: Can decode from $\geq \frac{n}{m+1} + \frac{m}{m+1} \cdot k$ non-errors provided $\alpha_1 \ldots \alpha_n$ distinct & order γ large.

Lemma ⇒ Theorem ?

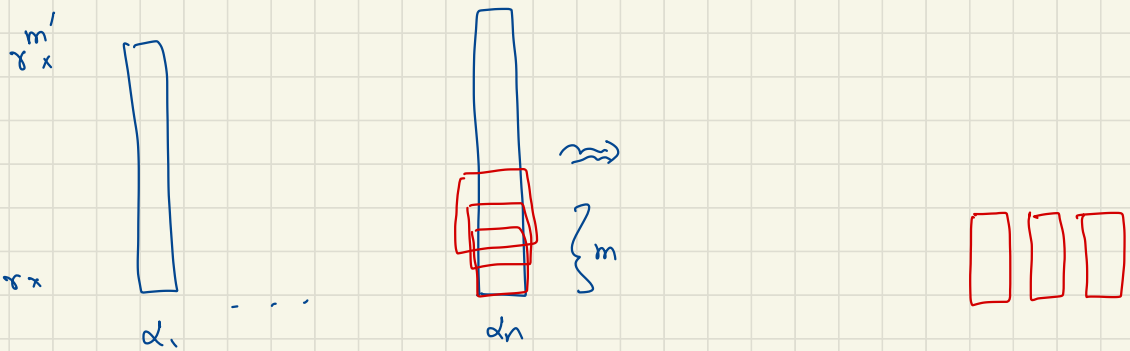
$$\text{Rate} = \frac{k}{mn}$$

$$\text{Correction} = \frac{1}{n}\left(\frac{n}{m+1} + \frac{m \cdot k}{m+1}\right) \approx 1 - m \cdot R + \delta n$$

$$[m = \tfrac{1}{8}]$$

Oops?

Fix: Start with $FRS_{n,k,r,m'=100 \cdot m}$



- Gives code with $n' = (99m)\,n$

  k = same

- Fraction of errors = same

- # non errors corrected $= \left(\frac{m \cdot k}{m+1} + \frac{n'}{m+1}\right)\frac{1}{n'} \approx (1.01)\frac{k}{m'n} + \frac{\delta n}{2}$

  $\approx k/n + \delta n$

Lemma: Can decode from $\geq \frac{n}{m+1} + \frac{m}{m+1} \cdot k$ non-errors provided $\alpha_1 \dots \alpha_n$ distinct & order γ large.

Problem: Given $k, n \dots$

$\gamma, \alpha_1 \dots \alpha_n$

$\beta_1^{(1)} \quad \beta_n^{(1)}$

$\vdots$

$\beta_1^{(m)} \quad \dots \quad \beta_n^{(m)}$

Output all p, $\deg p < k$, s.t.

$$\left| \{ i \mid \forall j \in [m] \quad p(\gamma^j \cdot \alpha_i) = \beta_i^{(j)} \} \right| \geq \underbrace{\frac{mk}{m+1} + \frac{n}{m+1}}_{D}$$

Algorithm:

I. Find $Q(X, Y_1 \dots Y_m) \not\equiv 0$ s.t. $Q(\alpha_i, \beta_i^{(1)} \dots \beta_i^{(m)}) = 0 \quad \forall i$

$Q(X, Y_1 \dots Y_m) = A_0(X) + Y_1 A_1(X) + \dots Y_m A_m(X)$

$\deg A_0 \leq D$; $\deg A_i \leq D - k$

II. Find all $p(x)$ s.t.

$$Q(X, p(X), p(\gamma X), \dots p(\gamma^m X)) \equiv 0$$

Analysis

1. Q as required in Step I exists

   Pf: # degrees of freedom $= D + m(D-k) > n$

2. Q can be found algorithmically

   Pf: linear system

3. $\forall Q$ found in Step 1, if P is a solution then

$$Q(x, P(x), P(\gamma x), \cdots P(\gamma^{m-1}x)) \equiv 0$$

   Pf: Let $R(x) \equiv Q(x, P(x), \ldots P(\gamma^{m-1}x))$

   (i) Deg $R \leq D$

   (ii) $\forall i$ s.t. $(P_i^{(1)} \cdots P_i^{(m)}) = (P(\alpha_i), P(\gamma\alpha_i) \cdots P(\gamma^{m-1}\alpha_i))$

$$R(\alpha_i) = 0$$

   (iii) $\Rightarrow R \equiv 0$. ⊠

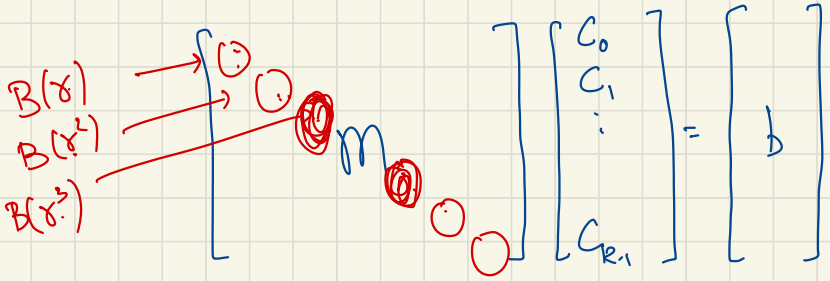
4. Can find all P s.t. $Q(x, P(x), \ldots P(\gamma^{m-1}x)) \equiv 0$

   Pf: Solve linear system! $P(x) = C_0 + C_1 x + \cdots C_{k-1} x^{k-1}$

5. # such P small? Dimension of linear system small?

High level view of Step 5 analysis

(i) Write linear system: $P(X) = C_0 + C_1 X + \dots C_{k-1} X^{k-1}$



(ii) M upper triangular — order$(\gamma) > k$, $\{\gamma^0, \gamma^1 \dots \gamma^{k-1}\}$ distinct

Captures condition $Q(X, P(X), \dots) \equiv 0 \pmod{X^i}$

(iii) Diagonal entries of M — No $f(x) \mid Q(X, Y_1, \dots Y_m)$

- of the form $B(\gamma^0), B(\gamma^1), \dots B(\gamma^*)$

  deg $B \le m$

- Can ensure $B \neq 0$

(iv) # solution $\le q^m$ ☒

Summary: have shown

- $\forall \delta > 0, \epsilon > 0$ $\forall$ suff large n $\exists q, L = \text{poly}(n)$ & code $E: \Sigma_q^k \to \Sigma_q^n$ with $\frac{k}{n} \geq 1 - \epsilon - \delta$ that is (ϵ, L)-list-decodable.

- Proof is algorithmic!

- Improvements:
  - $q = O_\delta(1)$: [Guruswami, Rudra] + [Guruswami-Indyk]
  - $L = O_\delta(1)$: [Kopparty, Ron-Zewi, Saraf, Wootters]

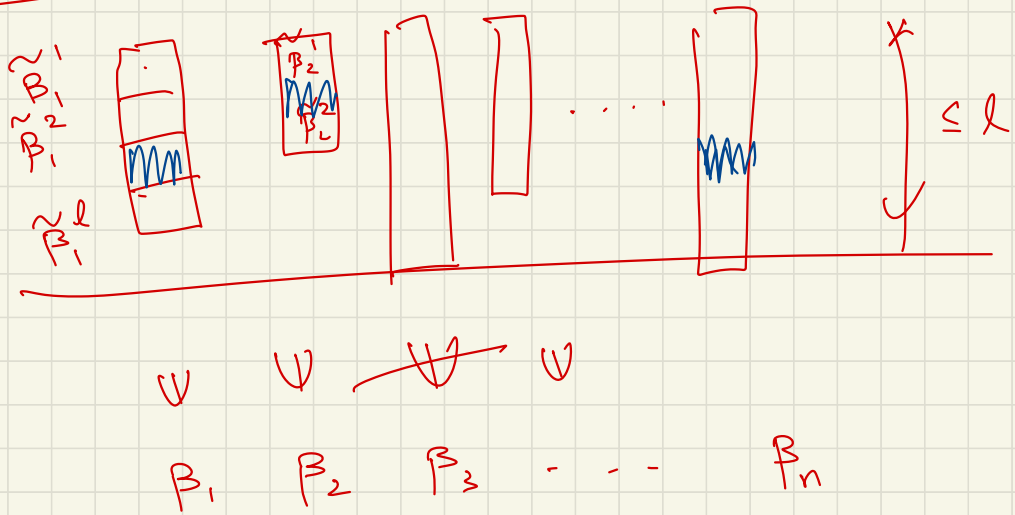
List-recoverability

$\beta_i \in \Sigma$

Channel

Usual List-Decoding

List-Recovery



As long as # errors in list-recovery sense

for every ℓ, can solve (ϵ, ℓ, L)-list-recovery with rate $1 - \epsilon - \delta$ alphabet grows with δ, ℓ.

Next lectures:

Local Coding Theory

- Context: if data is really large ... we can encode it all together & benefit from scale; but decoding time grows with data size. Is this necessary?
- Local Coding Theory: Explores algorithms that can recover any one bit of message with few randomized queries.

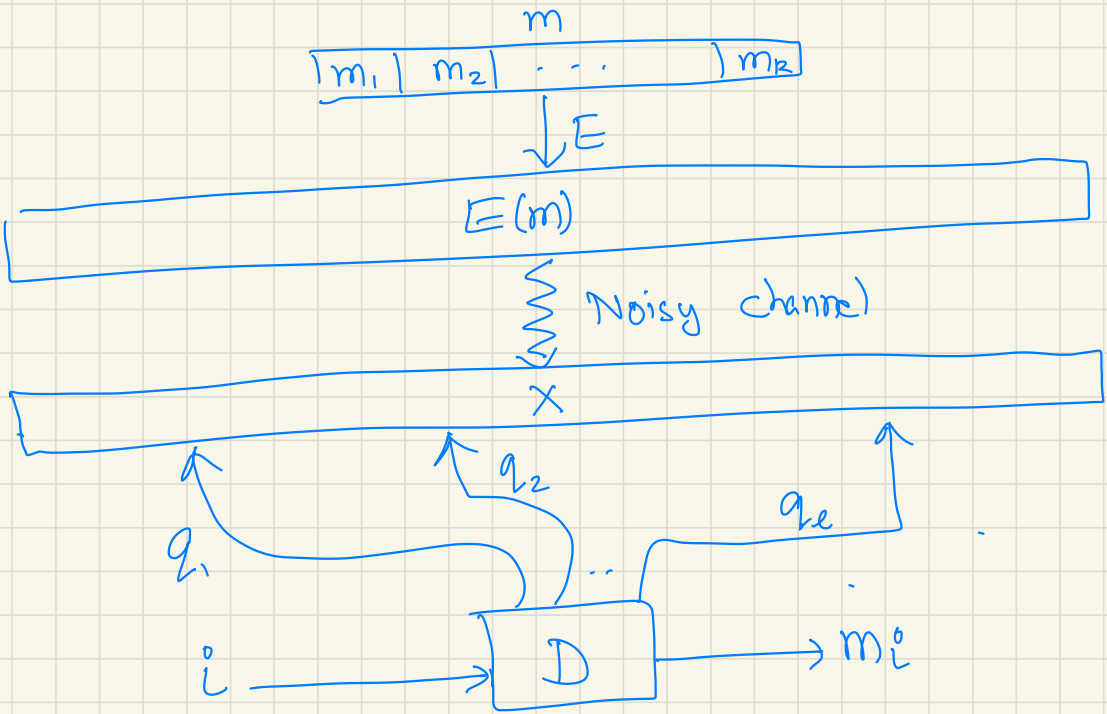
  # queries =: locality

Algebra In Algorithmic Coding Theory - 3

Today: Locally Decodable Codes

$E: \Sigma^R \to \Sigma^n$ is (ℓ, ϵ)-Locally Decodable (LDC) if $\exists$ Decoder D s.t. $\forall x$ s.t. $\Delta(x, E(m)) \leq \epsilon$

- $\forall i \in [R] \quad \Pr_D[D^x(i) = m_i] \geq \frac{2}{3}$
- D only makes ℓ queries to x.



(ℓ, ϵ)-LDC: corrects ϵ-fraction error with ℓ queries

A Classical Example: Hadamard Code

- $H: \mathbb{F}_2^k \to \mathbb{F}_2^{2^k}$

$$(a_1 \dots a_k) \longmapsto \langle L(x) \rangle_{x \in \mathbb{F}_2^k}$$

$$L(x_1 \dots x_k) = \sum_{i=1}^{k} a_i x_i$$

- Maps k bits $\to$ 2^k bits
- Claim: $(2, \frac{1}{4})$-LDC
  - Proof: Given received word $w: \mathbb{F}_2^k \to \mathbb{F}_2$
  - Decoder(i): Pick $r = (r_1 \dots r_k) \in \mathbb{F}_2^k$ u.a.r.

    Output $w(r+e_i) - w(r)$

    (where $e_i = (\underbrace{0\,0\,0}_{i-1}\,1\,\underbrace{0\,0\,0}_{k-i})$)
  - Correctness: Suppose $\Delta(w, E(a)) < \frac{1}{4} 2^k$

$$\Pr_r \left[w(r) \neq \sum_j a_j r_j \right] < \frac{1}{4}$$

$$\Pr_r \left[w(r+e_i) \neq a_i + \sum_j a_j r_j \right] < \frac{1}{4}$$

$$\Rightarrow \Pr_r \Big[\underbrace{w(r) = \textstyle\sum a_j r_j \;\wedge\; w(r+e_i) = \sum a_j r_j + a_i}_{> \frac{1}{2} \;\Rightarrow\; \text{output} = a_i} \Big]$$

A General Example: Reed Muller Codes

- Messages $\approx$ m-variate polynomials of degree r over $\mathbb{F}_q$
- Encoding $\approx$ Evaluation on all of $\mathbb{F}_q^m$.
- Formally: Let $T_{q,m,r} \subseteq \mathbb{F}_q^m$ be an interpolating set

  ie, a set with the property that for every function $f : T_{q,m,r} \to \mathbb{F}_q$ $\exists$ unique polynomial $P \in \mathbb{F}_q[x_1 \dots x_m]$ of total deg $\leq r$, ind. degree $< q$.

  $\forall \alpha \in T_{q,m,r}$ $\quad f(\alpha) = P(\alpha)$

- Message Space of $RM[q,m,r] = \{ f : T_{q,m,r} \to \mathbb{F}_q \}$

  $\cong \mathbb{F}_q^{|T_{q,m,r}|}$

- Encoding $E_{RM}(f) = P$ s.t. $P|_{T_{q,m,r}} = f$.
- Facts: if $r < q$ then $|T_{q,m,r}| = \binom{m+r}{r}$.

  if $r < q$ then $\dfrac{\Delta(P_1, P_2)}{q^m} \geq 1 - \dfrac{r}{q}$

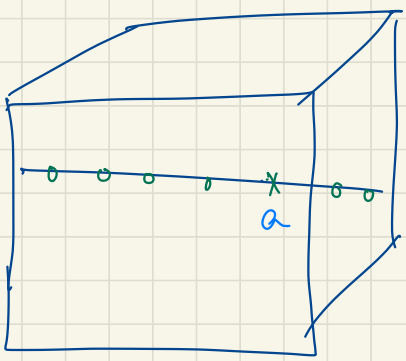
(Exercise to Prove this)

Local Decodability of RM Codes: $r < q$

Claim: RM codes decodable from $\Omega(1 - \frac{r}{q})$ fraction error with q locality

"$(q, \Omega(1-\frac{r}{q}))$" LDC

Proof: **Idea:**



- To compute $P(a)$:
- Pick random line ℓ containing a.
- Facts: $P|_\ell$ is a univ. deg r poly.
- Every point on ℓ (except a) is a uniform point in $\mathbb{F}_q^m$.
- Can recover $P|_\ell$ whp & interpolate to get $P(a)$.

Bit more formally

$$\left[\delta(f,P) \triangleq \frac{\Delta(f,P)}{q^m}\right]$$

PROBLEM: Given ① query access to $f: \mathbb{F}_q^m \to \mathbb{F}_q$ s.t. $\exists$ $P \in \mathbb{F}_q^{\leq r}[x_1 .. x_m]$ s.t. $\delta(f,P) \leq \epsilon$

② $a \in \mathbb{F}_q^m$

Output: $P(a)$ (with probability $> \frac{1}{2}$) ($\geq \frac{2}{3}$)

Algorithm $D^f(a)$: ① Pick $b \in \mathbb{F}_q^m$ uniformly $\boxed{\ell_{a,b} \triangleq \{a + t \cdot b \mid t \in \mathbb{F}_q\}}$ ← line

② Define $g(t) \triangleq f(a + t \cdot b)$ ← queries

③ Compute $h \in \mathbb{F}_q^{\leq r}[T]$ s.t. $\delta(g,h) \leq 3\epsilon$

④ Output $h(0)$

Correctness Claims: $P|_\ell(T) \triangleq P(a + T \cdot b)$

① $\Pr_b\left[\delta(g, P|_\ell) > 3\epsilon\right] \leq \frac{1}{3}$

② $\delta(g, P|_\ell) \leq 3\epsilon$ & $3\epsilon < \frac{1}{2}\left(1 - \frac{r}{q}\right) \Rightarrow P|_\ell = h$

③ $P|_\ell(0) = P(a)$

Proofs:

① + ② + ③ ⇒ Algorithm correct !!

Proof of ③: obvious by definitions: $P|_\ell(T) = P(a+Tb) \Rightarrow P|_\ell(\theta) = P(a+\theta b)$

Proof of ②: $\delta(P|_\ell, h) \le \delta(P|_\ell, g) + \delta(g, h)$

$\le \delta(P|_\ell, g) + \delta(g, P|_\ell)$ [h minimizes $\delta(g,h)$ among $\mathbb{F}_q^{\le r}[T]$]

$\le 6\epsilon < 1 - \frac{r}{q}$

But $P|_\ell, h \in \mathbb{F}_q^{\le r}[T] \Rightarrow \delta(P|_\ell, h) > 1 - \frac{r}{q}$ or $P|_\ell = h$ ☒

Proof of ①: $\forall \theta \in \mathbb{F}_q$; $a + \theta \cdot b \sim \text{Unif}(\mathbb{F}_q^m)$

$\Pr_b[f(a+\theta\cdot b) \neq P(a+\theta\cdot b)] \le \epsilon$.

$\Rightarrow \Pr_{\theta,b}[\quad \lesssim \quad] \le \epsilon$

$\Rightarrow \Pr_b\left[\Pr_\theta[f(a+\theta b) \neq P(a+\theta\cdot b)] \ge 3\epsilon\right] \le \frac{1}{3}$ [Averaging]

$\Leftrightarrow \Pr_b\left[\Pr_\theta[g(\theta) \neq P|_\ell(\theta)] \ge 3\epsilon\right] \le \frac{1}{3}$

$\Leftrightarrow \Pr_b[\delta(g, P|_\ell) \ge 3\epsilon] \le \frac{1}{3}$ ☒

Where we are ("were" in 2010)

- Can we $r = O(1)$ $\Rightarrow$ $m \to \infty$; $q = r+1$

$$k \approx m^r \Rightarrow m \approx k^{1/r}$$

$$n = q^m = \exp(k^{1/r}) \; ; \quad \ell = q-1 = r$$

$$\Rightarrow \boxed{n = \exp(k^{1/\ell}) \; ; \; \ell = O(1)}$$

- Can we $m = O(1)$ $\Rightarrow$ $r \to \infty$; $q = \frac{r}{1-\delta}$

$$k \approx \frac{r^m}{m!} \approx \frac{(1-\delta)^m}{m!} q^m$$

$$\ell \approx q \approx n^{1/m}$$

$$\boxed{n \approx k \cdot \frac{(1-m\delta)}{m!} \; ; \; \ell = n^{1/m} \approx m k^{1/m}}$$

- One more setting: $m = O\left(\frac{\log k}{\log\log k}\right)$; $r = O(\log^2 k)$

$$q = 2r$$

yields $\boxed{n \approx k^2 \; ; \; \ell = O(\log^2 k)}$

Bottomline: No $o(k)$ locality unless $\frac{k}{n} < \frac{1}{2}$.

2010 Breakthrough : [Kopparty Saraf Yekhanin]

"Multiplicity Codes" : will do special case to motivate.

- for Poly $P(X,Y) \in \mathbb{F}_q[X,Y]$, let $P_X = \frac{\partial P}{\partial X}$ & $P_Y = \frac{\partial P}{\partial Y}$
- ($m=2$, $s=2$ special case)
- messages : $P(X,Y) \in \mathbb{F}_q[X,Y]$, $\deg P \le r < 2q$
- Encoding : $P \mapsto \left(P(a,b), P_X(a,b), P_Y(a,b)\right)_{(a,b) \in \mathbb{F}_q^2}$.

  Code : $\mathbb{F}_q^{\binom{r+2}{2}} \longrightarrow \left(\mathbb{F}_q^3\right)^{q^2}$

- Distance $\Leftarrow$ Multiplicity Distance Lemma

  Def: (a,b) — zero of multiplicity $\underline{2}$ if $(P, P_X, P_Y)(a,b) = (0,0,0)$

  — zero of multiplicity 1 if $P(a,b)=0$. otherwise

$$\text{mult}_P(a,b) = 0 \text{ if } P(a,b) \neq 0$$
$$= i \text{ if } (a,b) \text{ is a zero of mult. } i \text{ of } P.$$

Multiplicity Distance Lemma: if $P \in \mathbb{F}_q[X_1 \dots X_m] \neq 0$ &
$\deg P \leq r$ then

$$\sum_{\bar{a} \in \mathbb{F}_q^m} \text{mult}_P(\bar{a}) \leq r \cdot q^{m-1} .$$

$$\Rightarrow \delta(\text{mult-code}_{m=2,s=2}) \geq 1 - \frac{r}{2q} \quad \left(1 - \frac{r}{sq}\right)$$

---

local-decoding: similar idea to before; Given f, f_x, f_y

recover $P(a,b)$:
- Pick $c,d \in \mathbb{F}_q$ n.a.r.
- $g(T) \triangleq f(a+cT, b+dT)$; $P|_\ell(T) \triangleq P(a+cT, b+dT)$
- $g'(T) \triangleq c \cdot f_x(a+cT, b+dT) + d \cdot f_y(a+cT, b+dT)$
- Find $h(T) \in \mathbb{F}_q^{\leq r}[T]$ that minimizes $\delta((h,h'),(g,g'))$.
- Output $h(0)$.

recovering $P_x(a,b)$, $P_y(a,b)$: Similar
... first find $c \cdot P_x(a,b) + d \cdot P_y(a,b)$
... then $c' \cdot P_x(a,b) + d' \cdot P_y(a,b)$
} combine.

Analysis: Similar to Reed-Muller + Derivatives.

Performance: Can set $r = (1-\delta)2q$. will set $\delta = 0$ to simplify expressions.

- $k = \binom{r+2}{2} \approx 2q^2$
- $n = 3 \cdot q^2 \Rightarrow \frac{k}{n} \to \frac{2}{3} > \frac{1}{2}$!
  (p, p_x, p_y)
- locality $\propto O(q) \propto O(\sqrt{k})$ ← Reed-Muller $1/2$

- Can improve further by taking higher order derivatives

$\Rightarrow R \triangleq \frac{k}{n} \to 1$; locality $\approx O(\sqrt{k})$

- Can improve even further by taking more variables.

$\Rightarrow R \to 1$; locality $\approx O(k^{\epsilon})$

Thm [KSY 2010]: $\forall \alpha, \beta > 0 \ \exists \delta > 0$ s.t. $\exists$ multiplicity codes with $R = \frac{k}{n} \geq 1-\alpha$; that are (n^{β}, δ)-LDC.

State of the art: [Kopparty, Meir, Ron-Zewi, Saraf]

$\forall\ R \in [0,1]$, $\forall\ \alpha > 0$ $\exists$ codes of Rate R, correctible from $\left(\frac{1-R-\alpha}{2}\right)$-fraction errors with locality

$$2^{\tilde{O}(\sqrt{\log n})}!$$

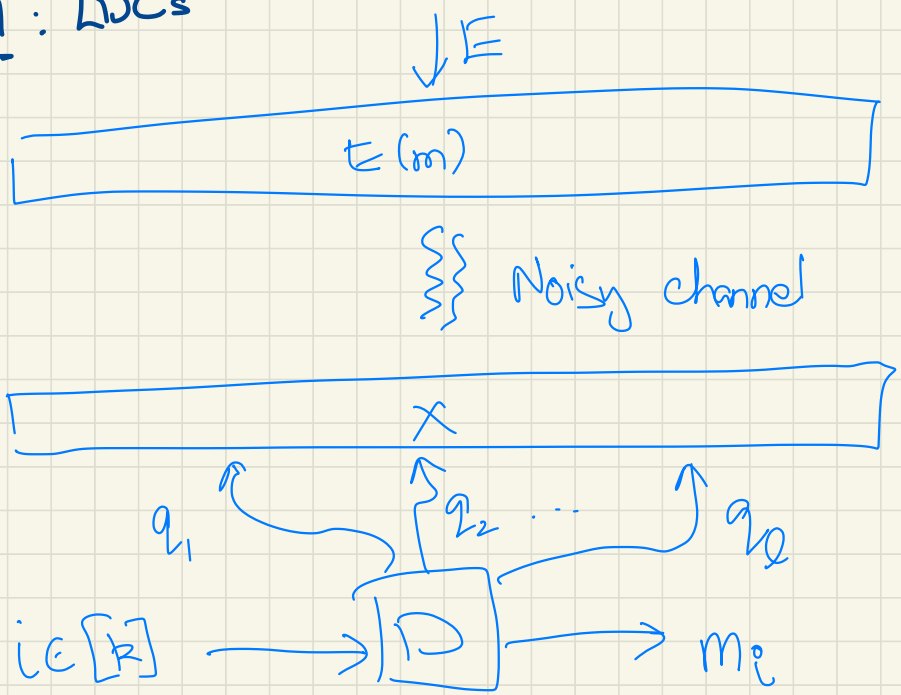
(Idea: More multiplicity + graph-theoretic amplification)

Next Lecture: $O(1)$-locality codes.

Algebra In Algorithmic Coding Theory - 3

Today: Local Codes with O(1) Locality & PIR Schemes.

Recall: LDCs



$E: \Sigma^k \to \Sigma^n$ is (ℓ, ϵ)-locally decodable code (LDC) if $\exists D$ s.t. $\forall x$ s.t. $\Delta(x, E(m)) \leq \epsilon \cdot n$, $\forall i$, $\Pr[D^x(i) = m_i] \geq \frac{2}{3}$ & D makes ℓ queries to x.

(ℓ, k, c) – PIR Scheme

- Database : $f : [k] \to \{0,1\}$
- ℓ - servers ;
- User(i): sends $\leq c$ bits to each server
  receives $\leq c$ bits from each server.
  recovers $f(i)$.
- $\forall$ server: distribution of message received ind of i
  [i is private]

LDC is uniform if $\forall i \in k$, each query of decoder $D(i)$ is marginally uniform on $[n]$

Facts: • LDCs we know are uniform

• Uniform 2DCs $\Rightarrow$ PIR schemes.

Suppose $E : \{0,1\}^k \to (\Sigma)^N$ is uniform $(\ell, s(\ell))$-LDC.

Then PIR scheme for $f : [k] \to \{0,1\}$ is as follows.

Given i : let $q_1 \ldots q_\ell$ be queries of $D(i)$.

Send $q_j \to$ server j ($\log N$ bits user → server)

Server sends $E(f)_{q_j} \to$ User ($\log \Sigma$ bits server → user)

Best known PIR schemes

- Based on RM codes: ℓ-server $\Rightarrow$ $K^{\frac{1}{2\ell}}$ communication
  + derivatives
- Some improvements: ℓ-server $\Rightarrow$ $K^{o(\frac{1}{\ell})}$
- [2006-2008] [Yekhanin - Efremenko]

  3-query LDC with $\exp(\sqrt{\log K}) = K^{o(1)}$ communication.

  ($\Rightarrow$ 3-server PIR)
- [2015] (Dvir-Gopi)

  2-server LDC with $\exp(\sqrt{\log K})$ comm.

TODAY: Proof via [Ghasemi, Kopparty, S. '24].

Key Ingredient: Matching Vectors

$u_1, \dots, u_K \quad v_1, \dots, v_K \in \mathbb{Z}_m^t$

form S-matching vectors for $S \subseteq \mathbb{Z}_m$ if

① $\forall\, i,j \quad \langle u_i, v_j \rangle \in S$

② $\langle u_i, v_j \rangle = 0 \iff i = j$

Example: ① $S = \{0,1\}$, m = prime

then $t \geq K - 1$ [matrix M with $M_{ij} = \langle u_i, v_j \rangle$ is almost full rank]

② S general, m = prime

$t \geq K^{1/m-1}$ [Exercise]

(Shocking) Theorem: [Beigel Barrington Rudich] [Grolmusz]

if $m = pq$ (distinct prime) then can make

$t \approx \exp\left(\tilde{O}\left(\sqrt{\log K}\right)\right)$

Matching Vectors $\Rightarrow$ LDCs

- Given message $f = f(1) \dots f(K)$

  & matching vectors $U(1) \dots U(K) \in \mathbb{Z}_m^t$

  $V(1) \dots V(K) \in \mathbb{Z}_m^t$

- Pick $\mathbb{F}_q$ so that it has m^{th} root of unity ω

- Notation: $X^{U(i)} \triangleq X_1^{U(i)_1} X_2^{U(i)_2} \dots X_t^{U(i)_t}$

  Given $\beta = (\beta_1 \dots \beta_t) \in \mathbb{Z}_m^t$ let $\omega^\beta = (\omega^{\beta_1}, \dots, \omega^{\beta_t})$

- Encoding of $f := \{F(\omega^\beta)\}_{\beta \in \mathbb{Z}_m^t}$ $\quad N = m^t$

  where $F(X) = F(X_1 \dots X_t) = \sum_{i=1}^{K} f(i) X^{U(i)}$

- Local Decoding to recover $f(i)$:

  - Pick $\beta = (\beta_1 \dots \beta_t)$ u.a.r $\in \mathbb{Z}_m^t$
  - Define $C_\beta(Y) = (\beta_1 Y^{V(i)_1}, \dots, \beta_t Y^{V(i)_t})$
  - Let $R(\Theta) = F(C_\beta(\Theta)) \quad \forall \Theta \in \{1, \omega, \dots \omega^{m-1}\}$
  - Interpolate degree $m-1$ poly $P(Y)$ s.t. $P(\Theta) = R(\Theta)$
  - Output 1 iff $P(0) \neq 0$.

Parameter Check:

- $N = m^t$; communication user → server $\log N$
  $\approx t$ ✓
  $\approx \exp(\sqrt{\log k})$

  server → user : $\log q$ bits $= O(1)$
  ✓✓

- # servers $= m = O(1)$ ✓
  [Not as good as Efremenko ...]
  [Can be improved to 3 using special properties of S in [BBR] & notion of S-interpolating set]

- Correct?
- 2-Server?

Correctness:

Main Idea: for simplicity fix $\beta = (0 \dots 0)$; so can ignore ω^β

(1) Define $G(Y) = F(C(Y))$

$$= \sum_{j=1}^{K} f(j) \cdot Y^{\langle U(j), V(i) \rangle}$$

← integer inner product

Note: deg G "high" $\approx \Theta_m(t)$

(2) But we plan to evaluate G on roots of unity.

So really care about

$$R(Y) = G(Y) \bmod (Y^m - 1)$$

$$= \sum_{j=1}^{K} f(j)\, Y^{\langle U(j), V(i) \rangle \bmod m}$$

↓ matching vector property

$$= f(1) \cdot Y^0 + \sum_{s \in S \setminus \{0\}} Y^s \cdot \underline{\text{some stuff}}.$$

- deg $R < m$; coeff of Y^0 is what we want!
- Can compute $R(\theta)$ $\forall\, \theta \in \{1, \omega, \dots \omega^{m-1}\}$

$\Rightarrow$ Can compute $R(Y)$ as polynomial!

[general β will scale coeff by some non-zero value, but preserves non-zero-ness.]

2-Servers?

- Idea: Use multiplicities. Say $m = 2 \cdot 3$

  Pick $\mathbb{F}_q$ of characteristic 3 (e.g. $\mathbb{F}_3$)

- Notice $(Y^6 - 1) = (Y^2 - 1)^3 = (Y-1)^3 (Y+1)^3$ in $\mathbb{F}_3[Y]$.

- First server asked for enough information as to compute $R(Y) \bmod (Y+1)^3$
- Second server ... $R(Y) \bmod (Y-1)^3$.

$\Rightarrow$ together $R(Y)$.

↳ How?: Ask for all first & second derivatives of $F(\,)$ at $C(1)$

Open Questions

- Can we get 2-server PIR with polylog$k$ comm.?
- Can we rule it out?