5708 LECTURE 16

Note Title 4/7/2008

TODAY

Conclude: [PARVARESH - VARDY]

[GURUSWAMI + RUDRA]

Rate-optimal list-decoding over large alphabets.

Review of last lecture

PARVARESH-VARDY CODES: Correlated RS Codes

Message: Degree $< k$ poly P_1 over $\mathbb{F}_q$

Alphabet: $\mathbb{F}_q^2$

Encoding: Let $P_2 = P_1^D \pmod{h(x)}$

Transmit $\{(P_1(\alpha_i), P_2(\alpha_i))\}_{i=1 \ldots n}$

Code Specs: $\mathbb{F}_q$, $\alpha_1 \ldots \alpha_n$, k, ← RS stuff

$D > \left(\frac{n}{k}\right)^{1/3}$, $h(x)$, monic, irred., deg. k — New stuff

Decoding Problem:

Given: Code Specs + $\{(\alpha_i, \beta_i, \gamma_i)\}_{i=1}^{n}$

Find: All poly P_1 s.t.

$$\left| \{ i \mid P_1(\alpha_i) = \beta_i \ \& \ P_2(\alpha_i) = \gamma_i \} \right| > t$$

for $P_2 = P_1^D \bmod h(x)$

Algorithm

Step 1: find $Q(x,y,z) \not\equiv 0$ s.t.

- $Q(\alpha_i, \beta_i, \gamma_i) = 0 \quad \forall i$
- $\deg_x Q \leq k^{2/3} n^{1/3}$
- $\deg_y Q, \deg_z Q \leq \left(\frac{n}{k}\right)^{1/3}$

Step 1.5 – while $h(x)$ divides $Q(x,y,z)$

$$Q \leftarrow Q/h \;;$$

Step 2: Let $Q_x^{(y,z)} = Q(x,y,z) \bmod h(x)$

Let $P_x(y) = Q_x(y, y^D)$

Output all roots of P_x in

$$E = \mathbb{F}_q[x]/h(x)$$

Analysis: (Won't repeat):

- P_1 has agreement $t > 3k^{2/3}n^{1/3}$
  $\Rightarrow$ P_1 root of $P_x(y)$.

- $P_x \not\equiv 0$, deg P_x not too large

Conclude:

- Get code of rate $R = \frac{k}{2n}$

- Corrects $\frac{n - 3k^{2/3}n^{1/3}}{n} = 1 - O(R^{2/3})$ fraction errors.

- Beats $1 - \sqrt{R}$ as $R \to 0$

Two Improvements

1. Multiplicities: Can use multiplicities trick to get rid of the '3' in $3 \cdot k^{2/3} n^{1/3}$

More precise result:

Code of rate $R = \frac{R'}{2}$

Correcting $1-(R')^{2/3}$ errors

$= 1-(2R)^{2/3}$ errors.

2. m-Correlated Polynomials

Can have $P_1 \dots P_m$

$$P_{i+1} = P_i^D \pmod{h(x)}$$

- Get code of rate R
  correcting $1-(mR)^{m/m+1}$ fraction errors
- Converges to $1-O\left(R\log\frac{1}{R}\right)$ fraction errors.
- Great for small rate!
  Big rate = ?
- [GURUSWAMI-RUDRA] Get rid of m in $1-(m\,R)^{m/m+1}$ by Algebraic Magic.

Idea: (Back in 2 polynomial setting)

- We're losing factor 2 in rate by using $P_1(x)$, & then $P_2(x)$.

- Could recover this if $P_2(x) = P_1(-x)$ for instance!

- Then we'd be transmitting

$$\begin{array}{|c|c|c|c|c|c|}
\hline
P_1(\alpha_1) & \cdots & P_1(\alpha_{n/2}) & P_1(-\alpha_1) & \cdots & P_1(-\alpha_{n/2}) \\
\hline
P_2(\alpha_1) & \cdots & P_2(\alpha_{n/2}) & P_2(-\alpha_1) & \cdots & P_2(-\alpha_{n/2}) \\
\hline
\end{array}$$

- But this is same as

$$\begin{array}{|c|c|c|c|c|c|}
\hline
P_1(\alpha_1) & \cdots & P_1(\alpha_{n/2}) & P_1(-\alpha_1) & \cdots & P_1(-\alpha_{n/2}) \\
P_1(-\alpha_1) & \cdots & P_1(-\alpha_{n/2}) & P_1(\alpha_1) & \cdots & P_1(\alpha_{n/2}) \\
\hline
\end{array}$$

where i^{th} symbol has same info as $\left(\frac{n}{2}+i\right)^{th}$ symbol.

- Don't need to send second half!
- Recover 2 in rate!!
- Can we implement our wish

$$P_2(x) \triangleq P_1(x)^D \pmod{h(x)} = P_1(-x) \pmod{h(x)}$$

for every P_1?

- Parameters under our control:

$\underline{h(x)}$, $\underline{D}$: can choose them as we please.

Some Basic Magic

Issue: How can we guarantee some identity of the form

$$P_1(x)^D = P_1(-x) \text{ for all } P_1 \,?$$

Insight: Using $D = q$ simplifies the quest...

$$P_1(x)^D = \left(\sum c_i x^i\right)^D$$

$$= \sum c_i^D x^{iD}$$

$$= \sum c_i x^{iD}$$

$$= P_1(x^D)$$

So suffices to have

$$X^{D} \equiv -X \pmod{h(x)}$$

$\Rightarrow$ need $h(x) \mid X^{D} + X$

BAD NEWS

Unfortunately ... this ties our hands.

$X^{q} + X$ is fixed. Can't have an irred. factor for most k.

[GR] Recovery:

1. Let's consider $X^{q} - \eta \cdot X$ for some $\eta \neq -1$.

2. Don't have to insist $h(x)$ of degree k; Magic takes care of degrees.

FACT: Let $\alpha \in \mathbb{F}_q^*$ be primitive.

- Then $X^{q-1} - \alpha$ is irreducible.
- Furthermore, $\forall\, P_1$

$$P_1(x)^q \equiv P_1(\alpha x) \pmod{X^{q-1} - \alpha}$$

- But we no longer have

$$\{P_1(x), P_2(x)\} \equiv \{P_1(\alpha x), P_2(\alpha x)\}$$

- Fix: Imperfect covers .....

GR Code:

in our example

Specs: $\mathbb{F}_q$, k, α, $(m = 2)$, $c = 3$

Message: P_1 of deg $< k$

Encoding:

$\boxed{P_1(\alpha), P_1(\alpha^2) \dots P_1(\alpha^c)}$ $\cdots\cdots$ $\boxed{P_1(\alpha^{q-c}) \dots P_1(\alpha^{q-1})}$

Decoding

$\boxed{y_1 \dots y_c}$ $\cdots\cdots$ $\boxed{y_{q-c} \dots y_{q-1}}$

$$\underbrace{\boxed{\begin{matrix} y_1 \\ \vdots \\ y_m \end{matrix}} \quad \boxed{\begin{matrix} y_2 \\ \vdots \\ y_{m+1} \end{matrix}} \quad \cdots \quad \boxed{\begin{matrix} y_{c-m+1} \\ \vdots \\ y_c \end{matrix}}}_{}$$

PV decode this with $D = q$, $h(x) = x^{q-1} - \alpha$

Performance

Rate $R = \dfrac{k}{q-1}$

Error correction: suppose t out of $\dfrac{q-1}{c}$ agreements

- transfers to $(c-m+1)t = t'$ out of $n' = (c-m+1)\left(\dfrac{q-1}{c}\right)$ symbols of PV code in agreement.

- PV decoder needs

$$t' > n' \cdot \left(\frac{k}{n'}\right)^{\frac{m}{m+1}}$$ agreements

- $R = \frac{k}{q-1}$; $q-1 = \frac{c \cdot n'}{c-m+1}$

$$\Rightarrow \quad \frac{k}{n'} = \frac{(q-1)\cdot R}{n'} = \frac{c \cdot R}{c-m+1}$$

- [GR] Code is decodable from

$$\left(\left(\frac{c}{c-m+1}\right)\cdot R\right)^{\frac{m}{m+1}}$$ fraction agreements

- Let $m = \frac{1}{\epsilon}$ & $c = \frac{1}{\epsilon^2}$

Then error $= 1 - (1+\epsilon)\cdot R^{(1-\epsilon)}$

$$\longrightarrow 1 - R - f(\epsilon)$$

Conclusions:

- Have found essentially best possible code + decoder over large alphabets.
- Downsides: Runtime = $\mathrm{poly}(n^{1/\varepsilon^2})$ ... not so nice. Can we do better?
- Questions: - Can RS codes be decoded better?
  - What other forms of folding would work well?