6.975 Graduate Seminar in Communications - Week 2

List-Decoding of Reed-Solomon Codes

Seminar Leader: Huan Yao

September 18, 2002

main paper : by Venkatesan Guruswami and Madhu Sudan

Improved decoding of Reed-Solomon and algebraic-geometry codes

$\boxed{\text{Outline}}$

- Motivation - Why List-Decoding
- Result Preview - How well does it work
- Johnson Bound - What is the limit
- The algorithm - How and Why
- Extensions to other codes

Notation

Code $(n, k+1, d)_q$

$$\begin{aligned} \mathbb{F}_q : &\quad \text{alphabet field} \\ n : &\quad \text{codeword length} \\ k+1 : &\quad \text{message length} \\ d : &\quad \text{min. Hamming distance} \end{aligned}$$

Hamming Distance :

$$\begin{aligned} \Delta(\mathbf{x}, \mathbf{y}) \quad = &\quad \text{number of coordinates} \\ &\quad \text{on which } \mathbf{x} \text{ and } \mathbf{y} \text{ differ} \end{aligned}$$

Hamming Ball : center $\mathbf{y}$, radius e

$$\mathrm{B}(\mathbf{y}, e) \stackrel{\text{def}}{=} \{\mathbf{x} \in \mathbb{F}_q^n | \Delta(\mathbf{x}, \mathbf{y}) \leq e\}$$

$$\mathrm{Vol_q}(e, n) = \sum_{i=0}^{e} \binom{n}{i} (q-1)^i$$

List-Decoding

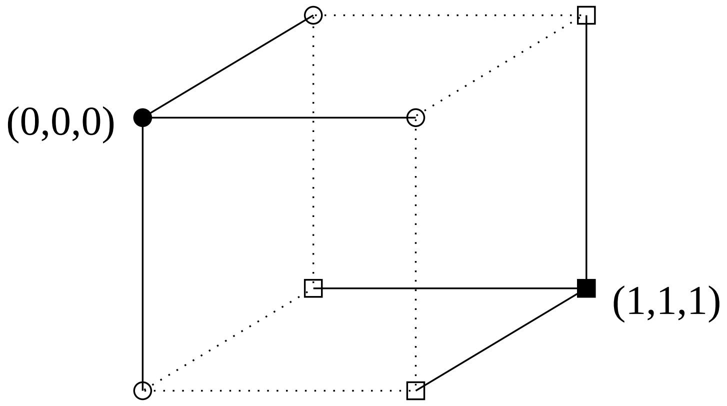
- **List-Decoding :** *The problem of finding all codewords within the Hamming ball around a word* $\mathbf{y}$ *with radius* e.
- **None, one, or more** codewords might be listed.
- **Error Event :** The true codeword not being listed.

Perfect Codes

meet *Hamming Bound*

$$q^{k+1}\mathrm{Vol}_q\left(\left\lfloor\frac{d-1}{2}\right\rfloor, n\right) \leq q^n$$

example 1: $(3,1,3)_2$ code



example 2: $(7,4,3)_2$ code

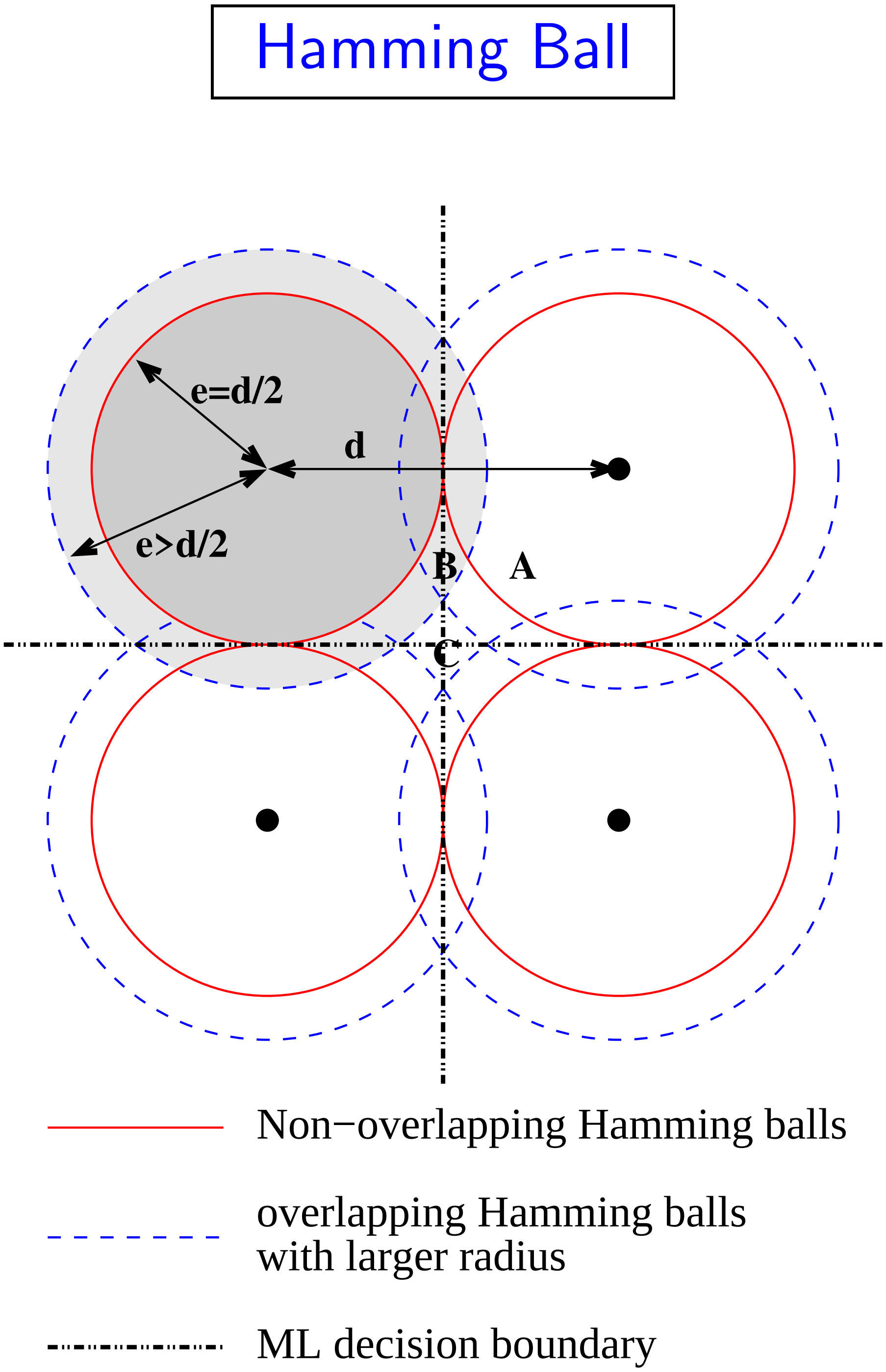
$$2^4 \cdot \mathrm{Vol}_2\left(\left\lfloor\frac{3-1}{2}\right\rfloor, 7\right) = 2^4 \cdot (1+7) = 2^7$$

Maximum Distance Separable (MDS) Codes

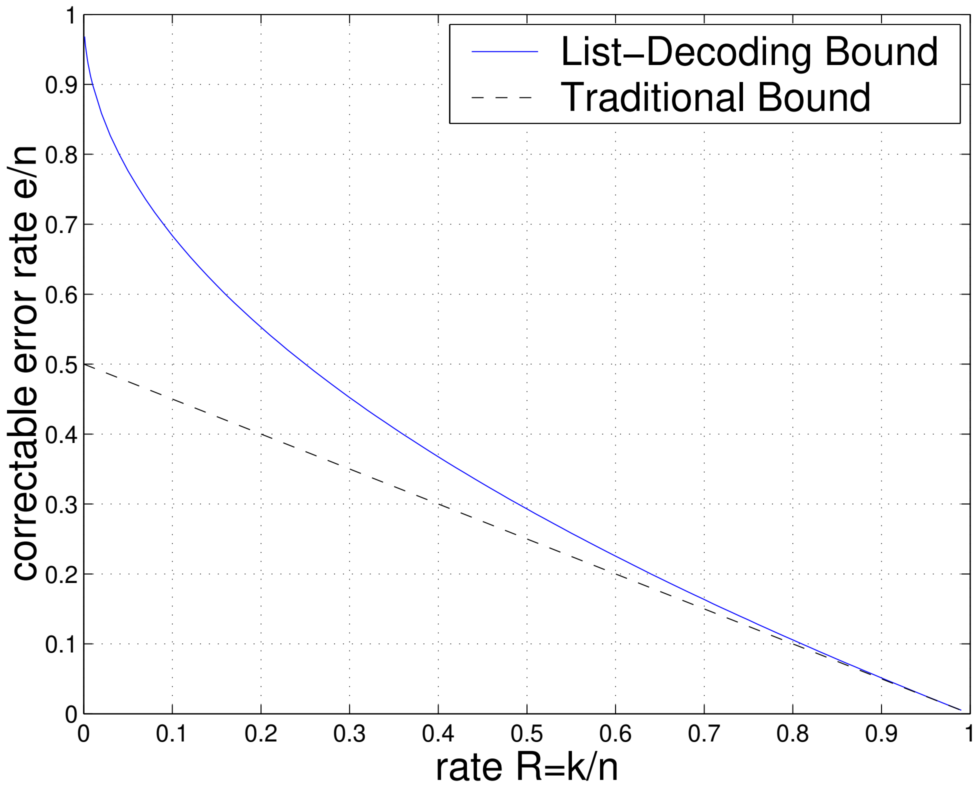
meet *Singleton Bound*

$$d \leq n - k$$

example : All Reed-Solomon Codes


Hamming Ball
e=d/2
d
e>d/2
B
A
C
Non−overlapping Hamming balls
overlapping Hamming balls
with larger radius
ML decision boundary

Comparison of error correction capability



$$\text{List-Decoding}: \quad e/n < 1 - \sqrt{k/n}$$

$$\text{Traditional}: \quad e/n < (1 - k/n)/2$$

Numerical Examples

code	rate	e	SER	Block ER
$(64, 48, 17)_{64}$	$3/4$	8	0.029	1.0×10^{-4}
		9		1.5×10^{-5}
$(64, 32, 33)_{64}$	$1/2$	16	0.098	1.0×10^{-4}
		19		1.7×10^{-6}
$(64, 16, 49)_{64}$	$1/4$	24	0.185	1.0×10^{-4}
		33		5.4×10^{-10}
$(255, 239, 17)_{256}$	0.94	8	0.007	1.0×10^{-4}
		8		1.0×10^{-4}
$(256, 128, 129)_{256}$	$1/2$	64	0.161	1.0×10^{-4}
		75		4.5×10^{-8}

Johnson Bound

$A'_q(n,d,e) \stackrel{\text{def}}{=}$ the maximum number of codewords distance d apart that are inside a Hamming ball with radius e.

$$\text{If} \quad e < \left(1 - \frac{1}{q}\right)\left(1 - \sqrt{1 - \frac{q}{q-1} \cdot \frac{d}{n}}\right) n,$$

$$\text{then} \quad A'_q(n,d,e) \leq \min\left\{ n(q-1), \frac{nd}{nd - 2e(n - \frac{qe}{2(q-1)})} \right\}.$$

Assume $q \approx q-1$, then the bound becomes:

$$\text{If} \quad e < n - \sqrt{n(n-d)} \quad \text{or} \quad \frac{e}{n} < 1 - \sqrt{\frac{k}{n}},$$

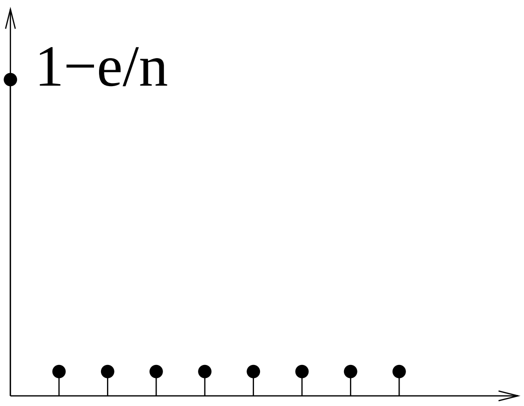
$$\text{then} \quad A'_q(n,d,e) \leq \min\left\{ nq, \frac{d}{d - (2e - e^2/n)} \right\}.$$

If we are over Johnson bound ...

If $e > (1+\epsilon)(n - \sqrt{n(n-d)}), \epsilon > 0$, *then there are exponential number of codewords in the Hamming ball.*

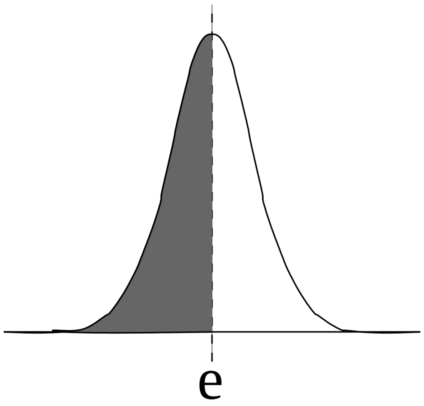
Consider Hamming ball around $\mathbf{0}$.

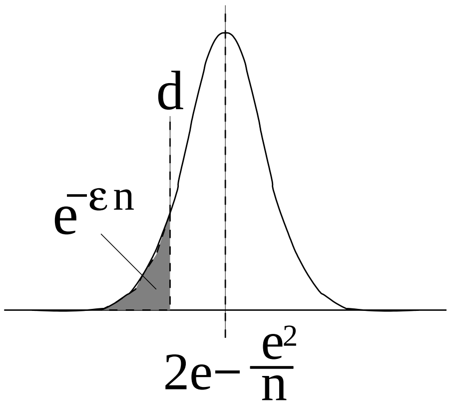
Pick random codewords with random coordinates according to distribution



Expected Hamming weight $= e$

Expected pairwise distance $n(1-(1-e/n)^2) = 2e - e^2/n$





A Brief History

- 1957, Elias, first introduced list-decoding.
- 1960, Peterson, polynomial time algorithm, $e \leq \lfloor (d-1)/2 \rfloor$.
- 1960's, Berlekamp and Massey, run time $O(n^2)$, same e.
- 1966, Forney, generalized minimum distance decoding, soft list-decoding.
- 1986, Welch and Berlekamp, univariate functions, $e \leq \lfloor (d-1)/2 \rfloor + 1$.
- 1989 and 1994, Dumer and Sidelnikov, $e < \lfloor (d-1)/2 \rfloor + c \log n$.
- 1997, Sudan, polynomial reconstruction, $e < n - \sqrt{2n(n-d)}$.
- 1999, Guruswami and Sudan, reaches $e < n - \sqrt{n(n-d)}$.
- 2000, Koetter and Vardy, soft decoding.

RS Encoding

- Pick distinct elements $x_1, x_2, \cdots, x_n \in \mathbb{F}_q$
- Given message $m_0, m_1, m_2, \cdots, m_k \in \mathbb{F}_q$
- Construct polynomial $p(x) \stackrel{\text{def}}{=} m_0 + m_1 x + m_2 x^2 + \cdots + m_k x^k$
- The output codeword is $(p(x_1), p(x_2), \cdots, p(x_n))$.

The Polynomial Reconstruction Problem

INPUT : *Integers k, t and n points $\{(x_i, y_i)\}_{i=1}^{n}$ where $x_i, y_i \in \mathbb{F}_q$.*

OUTPUT : *All polynomials $p(x)$ of degree at most k such that $y_i = p(x_i)$ for at least $t = n - e$ values of $i \in \{1, 2, \cdots, n\}$.*

Some Definitions

- **Singularity:** The meaning of $Q(x, y)$ having singularity r at (x_i, y_i) is the shift of $Q(x, y)$ to (x_i, y_i), i.e, $Q^{(i)}(x, y) = Q(x + x_i, y + y_i)$ has no monomial terms of degree less than r.

$$q^{(i)}_{j_1 j_2} \stackrel{\text{def}}{=} \sum_{j'_1 \geq j_1} \sum_{j'_2 \geq j_2} \begin{pmatrix} j'_1 \\ j_1 \end{pmatrix} \begin{pmatrix} j'_2 \\ j_2 \end{pmatrix} q_{j'_1 j'_2} x_i^{j'_1 - j_1} y_i^{j'_2 - j_2} = 0.$$

- **Weighted Degree:** For non-negative weights w_1 and w_2, the (w_1, w_2)-*weighted degree* of the monomial $x^i y^j$ is defined as $iw_1 + jw_2$.

The Algorithm

Input : *Integers* k, t *and* n *points* $\{(x_i, y_i)\}_{i=1}^{n}$ *where* $x_i, y_i \in \mathbb{F}_q$.

Step 0 : Compute parameters r, l such that

$$rt > l \text{ and } n \cdot \left(\frac{r(r+1)}{2}\right) < \frac{l(l+2)}{2k}.$$

Step 1 : Find a non-trivial polynomial $Q(x, y)$ with $(1, k)$-weighted degree at most l that has singularity r at all points $\{(x_i, y_i)\}_{i=1}^{n}$.

Step 2a : Find all polynomial $p(x)$ with coefficients in $\mathbb{F}_q$ of degree at most k such that $y - p(x)$ is a factor of $Q(x, y)$.

Step 2b : For each such polynomial $p(x)$, check if $p(x_i) = y_i$ for at least t values of $i \in 1, 2, \cdots n$, and if so, include $p(x)$ in output list.

How to get $y - p(x)$ to factor $Q(x, y)$

Lemma 4 *If* (x_i, y_i) *is an input point and* $p(x)$ *is any polynomial such that* $y_i = p(x_i)$*, then* $(x - x_i)^r$ *divides* $g(x) \stackrel{\text{def}}{=} Q(x, p(x))$*.*

$$(x - x_i)^r \mid Q\left(x, p(x)\right)$$

Lemma 5 *If* $p(x)$ *is a polynomial of degree at most* k*, such that* $y_i = p(x_i)$ *for at least* t *values of* $i \in \{1, 2, \cdots, n\}$ *and* $rt > l$*, then* $y - p(x)$ *divides* Q*.*

$$(x - x_1)^r (x - x_2)^r \cdots (x - x_t)^r \mid Q(x, p(x))$$

$$rt > l \Longrightarrow Q(x, p(x)) \equiv 0$$

Get the numbers to work out

Lemma 6 *If* $n \cdot \left(\frac{r(r+1)}{2}\right) < \frac{l(l+2)}{2k}$*, then a polynomial* $Q(x,y)$ *as sought in Step 1 does exist (and can be found in polynomial time by solving a linear system.)*

$$\forall i \in 1,2,\cdots n, \forall j_1, j_2 \geq 0, s.t., j_1 + j_2 < r,$$

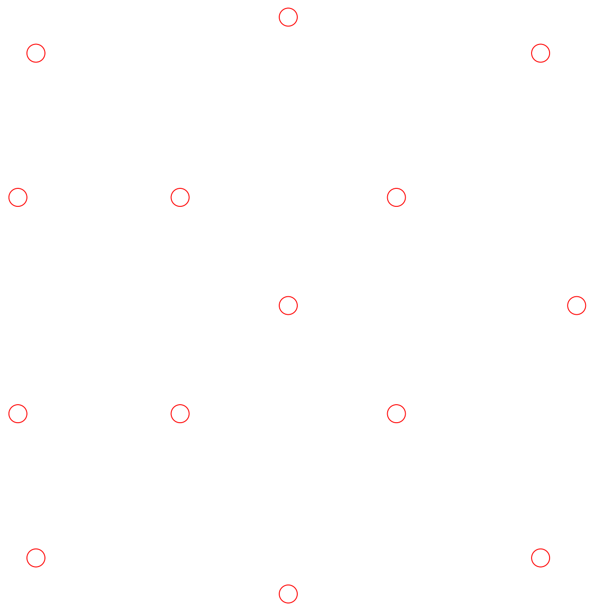
$$q^{(i)}_{j_1 j_2} \stackrel{\text{def}}{=} \sum_{j_1' \geq j_1} \sum_{j_2' \geq j_2} \left(\begin{array}{c} j_1' \\ j_1 \end{array}\right) \left(\begin{array}{c} j_2' \\ j_2 \end{array}\right) q_{j_1' j_2'} x_i^{j_1' - j_1} y_i^{j_2' - j_2} = 0.$$

Lemma 7 *If* n, k, t *satisfies* $t^2 > kn$*, then both conditions in Step 0 hold.*

$$\begin{aligned} r &\stackrel{\text{def}}{=} 1 + \left\lfloor \frac{kn + \sqrt{k^2n^2 + 4(t^2 - kn)}}{2(t^2 - kn)} \right\rfloor \\ l &\stackrel{\text{def}}{=} rt - 1 \end{aligned}$$

Example 1, setup

Example 1: $n = 14$, $k = 1$, and $t = 5$. ($\mathbb{F} = \mathbb{R}$)
find lines that pass through 5 points.

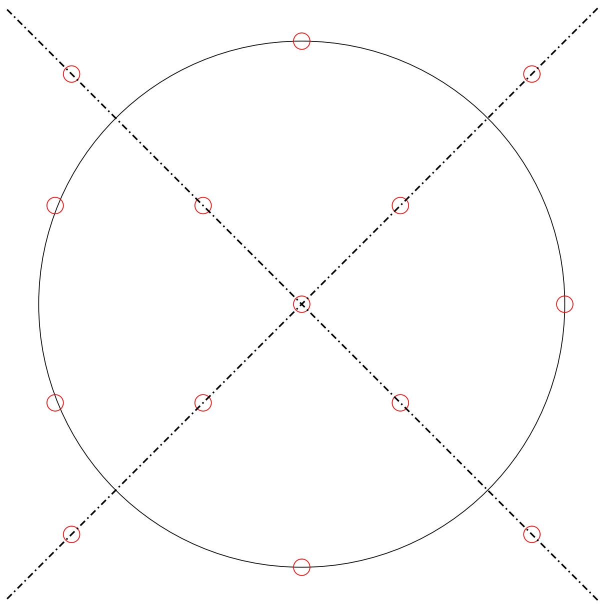


Example 1, solution

Example 1: $n = 14$, $k = 1$, and $t = 5$.

$$\text{Pick } r = 1 \longrightarrow l = 4$$
$$(14 \text{ equations, } 15 \text{ variables, } rt = 5 > 4 = l)$$

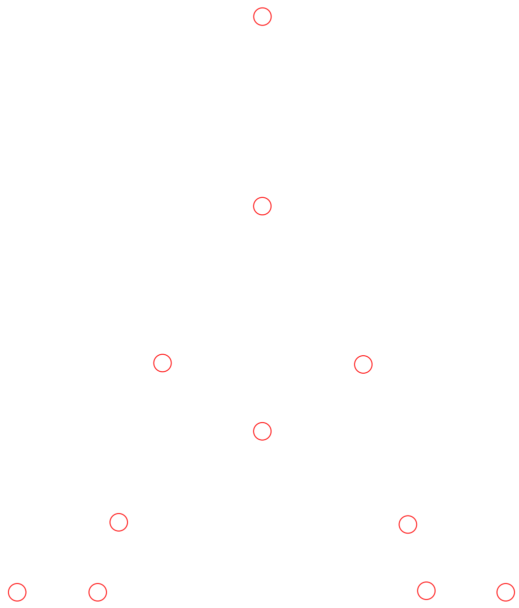
$$\begin{aligned} Q(x,y) &= y^4 - x^4 - y^2 + x^2 \\ &= (x^2 + y^2 - 1)(y + x)(y - x) \end{aligned}$$



Example 2, setup

Example 1: $n = 11$, $k = 1$, and $t = 4$. $(\mathbb{F} = \mathbb{R})$

find lines that pass through 4 points.

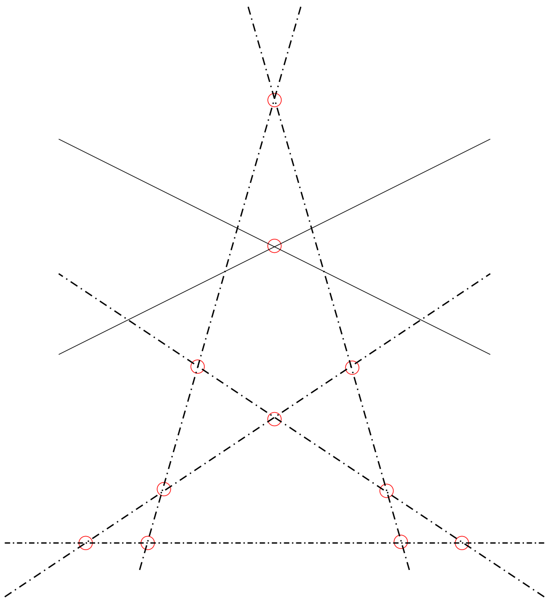


Example 2, solution

Example 2: $n = 11$, $k = 1$, and $t = 4$.

$$\text{Pick } r = 2 \longrightarrow l = 7$$

$$(33 \text{ equations, } 36 \text{ variables, } rt = 8 > 7 = l)$$



using $r = 1$ and $l = 4$ would not work

Complexity

Maximum Number of Solution :

$$M \leq \left\lfloor \frac{l}{k} \right\rfloor$$

Run Time:

Step 1 has $O(l^2/k)$ variables, takes $O((l^2/k)^3) = O(l^6/k^3)$ operations.

Step 2 takes $O(l^5/k^3)$ operations.

$$l = O(\max\left\{\frac{knt}{t^2 - kn}, t\right\}),$$

$$\text{run time} = O(\max\{\frac{k^3 n^6 t^6}{(t^2 - kn)^6}, \frac{t^6}{k^3}\}).$$

Numerical Examples

code	e	t	$t^2 - kn$	r	l	equations	max sol.
$(64, 48, 17)_{64}$	9	55	17	25	1374	20800	29
$(64, 32, 33)_{64}$	19	45	41	14	629	6720	20
$(64, 16, 49)_{64}$	33	31	1	495	15344	7856640	1022
	32	32	64	8	255	2304	17
$(256, 128, 129)_{256}$	75	181	249	39	7058	199680	55
	74	182	612	15	2729	30720	21
	73	183	977	10	1829	14080	14

Summary

- List-decoding can decode up to $e < n - \sqrt{n(n-d)}$.
- RS-decoding $\Longleftrightarrow$ Polynomial reconstruction problem

Extensions

- Can also be applied to alternant codes, AG codes, RM codes, etc.
- Soft version for erasure and AWGN channel and concatenated codes.