

Coding Theory

Tutorial

Brett Hemenway

UM Coding Complexity and Sparsity Workshop

August 1, 2011

Outline

Error Correcting Codes

List Decodable Codes

Locally-Decodable Codes

Matching Vector Codes

Multiplicity Codes

Error Correcting Codes

Definitions

- ▶ An error correcting code is a mapping

$$C : \mathbb{F}_q^k \rightarrow \mathbb{F}_q^n$$

- ▶ **Block Length:** n is called the block length of the code.
- ▶ **Rate:** k/n is called the rate of the code.
- ▶ **Minimum Distance:** The minimum hamming distance between two distinct codewords is called the minimum distance and is denoted d .
- ▶ **Relative Distance:** The relative distance, denote by δ , is the minimum distance divided by the block length, $\delta = d/n$.

Linear Codes

Parity Check Matrices

- ▶ Let G be an $n \times k$ (full rank) generator matrix of a code
- ▶ The (full rank) $n \times (n - k)$ matrix H with $HG = 0$ is called the *parity-check* matrix
- ▶ This means $\ker(H) = \text{im}(G)$
- ▶ Notice that $H(c + e) = Hc + He = He$ for any codeword c
- ▶ The vector He is called the syndrome
- ▶ Recovering e from He is sufficient for decoding

Linear Codes and Compressive Sensing

Connections

Linear Codes

Compressive Sensing

Linear Codes and Compressive Sensing

Connections

Linear Codes

Recover c from $c + e$

Compressive Sensing

Recover x from Φx

Linear Codes and Compressive Sensing

Connections

Linear Codes

Recover e from He

Compressive Sensing

Recover x from Φx

Linear Codes and Compressive Sensing

Connections

Linear Codes

Recover e from He

Compressive Sensing

Recover x from Φx

Both e and x are sparse!
Both H and Φ are short and fat!

Linear Codes and Compressive Sensing

Connections

Linear Codes

Recover e from He

Arithmetic is in \mathbb{F}_q

Compressive Sensing

Recover x from Φx

Arithmetic is in \mathbb{R}

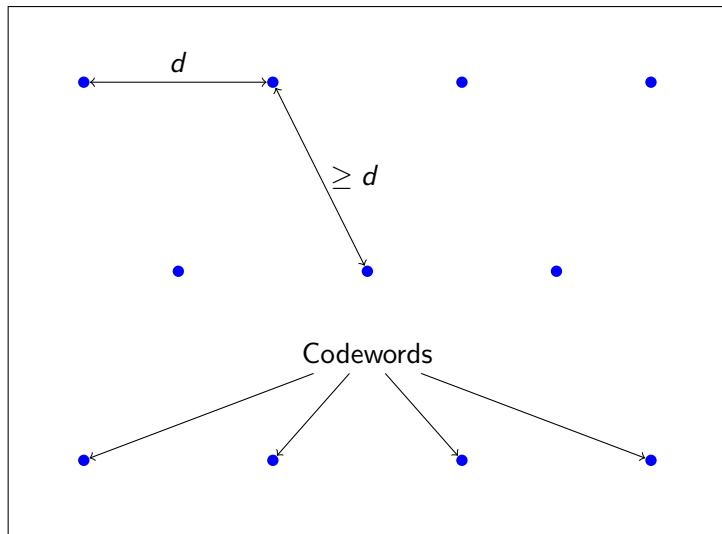
Connections Between Coding and Compressive Sensing

- ▶ Both seek to solve an underdetermined system for a sparse vector
- ▶ Model of arithmetic is different
- ▶ Compressive Sensing yields good codes (over \mathbb{R}) [CRTV05]
- ▶ A “good” binary parity-check matrix is a “good” measurement matrix [DV09]

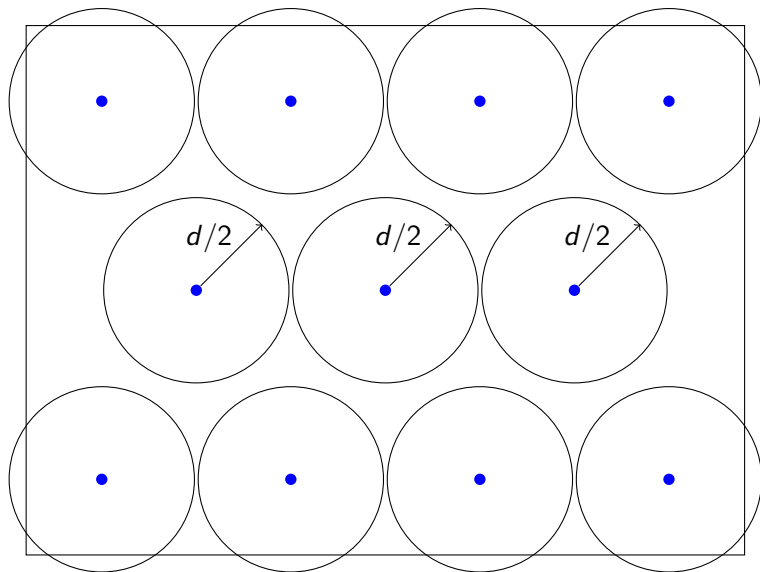
Basic Error Correction Bounds

- ▶ You can decode uniquely up to an error rate of half the relative distance (existentially)
- ▶ (Singleton Bound) The rate $R < 1 - \delta$
(Encoding is still injective if a δ fraction of the codeword is removed)
- ▶ Equivalently $\delta < 1 - R$
- ▶ Maximum tolerable error rate is thus $(1 - R)/2$

Packing Bounds for Codes



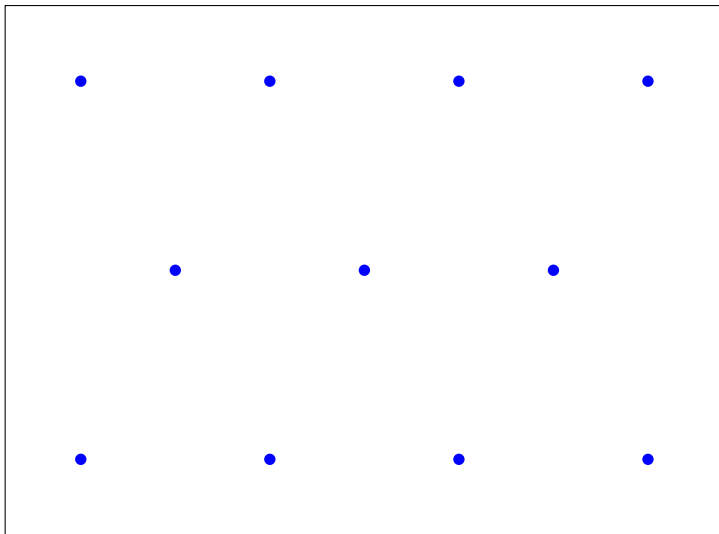
Packing Bounds for Codes



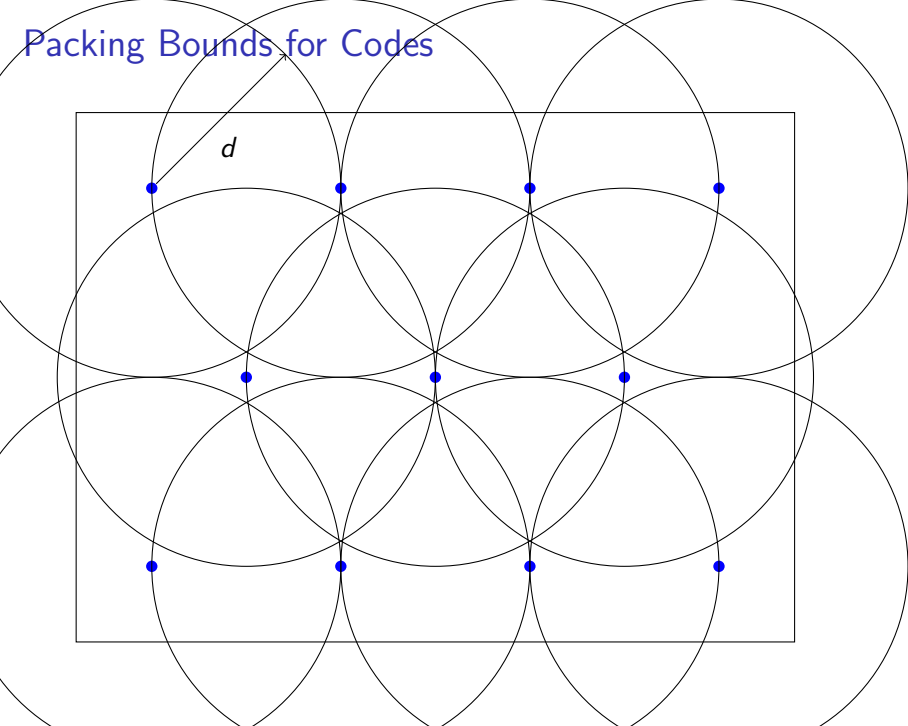
Sphere Packing Bound:

$$\# \text{ Codewords} \leq \frac{\text{Size of space}}{\text{Size of ball}} = \frac{q^n}{V_q(n, d/2)}$$

Packing Bounds for Codes



Packing Bounds for Codes



Gilbert Varshamov Bound:

$$\# \text{ Codewords} \geq \frac{q^n}{V_q(n,d)}$$

Packing Bounds for Codes

► **Volume of a ball:**

Let $V_q(n, d) = \sum_{i=0}^d \binom{n}{i} (q-1)^i$ be the volume of a ball of radius d in \mathbb{F}_q^n

► **Shannon Entropy:**

Let $H_q(\delta) = \delta \log_q(q-1) - \delta \log_q(\delta) - (1-\delta) \log_q(1-\delta)$
 $H_2(\delta) = -\delta \log_2(\delta) - (1-\delta) \log_2(1-\delta)$.

► Estimating $V_q(n, d) = q^{nH_q(d/n)} = q^{nH_q(\delta)}$

► Denoting the number of codewords q^k , we obtain

$$\frac{q^n}{2^{nH_q(\delta)}} \leq q^k \leq \frac{q^n}{2^{nH_q(\delta/2)}}$$

↓

$$\frac{q}{2^{H_q(\delta)}} \leq q^{k/n} \leq \frac{q}{2^{H_q(\delta/2)}}$$

↓

$$1 - H_q(\delta) \leq R \leq 1 - H_q(\delta/2)$$

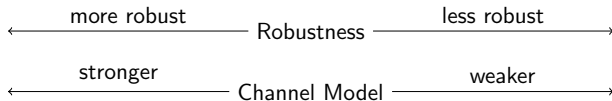
Modeling Errors

← stronger Channel Model weaker →

Hamming Model

- ▶ Geometric condition
- ▶ Worst case
- ▶ Guaranteed recovery
- ▶ Introduced by Hamming

Modeling Errors



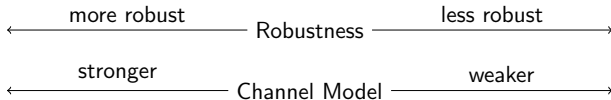
Hamming Model

- ▶ Geometric condition
- ▶ Worst case
- ▶ Guaranteed recovery
- ▶ Introduced by Hamming

Shannon Model

- ▶ Random errors
- ▶ Average case
- ▶ Probabilistic recovery
- ▶ Introduced by Shannon

Modeling Errors



Hamming Model

- ▶ Geometric condition
- ▶ Worst case
- ▶ Guaranteed recovery
- ▶ Introduced by Hamming

Shannon Model

- ▶ Random errors
- ▶ Average case
- ▶ Probabilistic recovery
- ▶ Introduced by Shannon

Bounded Channel Model

- ▶ Computational condition
- ▶ Worst case
- ▶ Probabilistic recovery
- ▶ Introduced by Lipton

The Bounded Channel Model

Overview of Results

- ▶ Introduced by Lipton [Lip94, GLD04]
A shared secret key transforms worst-case error into random error (code-scrambling)
- ▶ Digital signature schemes and list decoding allow decoding past unique decoding bound (PK model) [MPSW05]
- ▶ Shared secret key allows local decoding with constant rate [OPS07]
- ▶ Local decoding with constant rate (PK model) [HO08, HOSW11]
- ▶ Optimal rate codes against log-space bounded channels [GS10]

Another Connection to Compressive Sensing

Compressive Sensing for Simple Sources

- ▶ Computationally simple channels have proven useful in coding theory
- ▶ Exploiting the connection between codes and sparse approximation we can ask:
Can we make better measurement matrices for “simple” signals?
- ▶ In joint work with Anna, Atri, Martin and Mary we have results in this model

Reed Solomon Codes

- ▶ **Message:**

$f \in \mathbb{F}_q[z]$ of degree $k - 1$

- ▶ **Codeword:**

$(f(\alpha_1), \dots, f(\alpha_q))$

The evaluation of f at all points in \mathbb{F}_q .

- ▶ **Minimum Distance:**

The minimum distance of this code is $q - k + 1$.

If $f \neq g$, then $f - g$ has at most $k - 1$ zeros. Thus $C(f)$ and $C(g)$ differ in $q - k + 1$ coordinates.

- ▶ **Rate:**

$R = k/q$, and $\delta = (q - k + 1)/q$, so $R \approx 1 - \delta$

This matches the singleton bound

- ▶ Notice that the length of the code is bounded by the alphabet size.

Decoding Reed Solomon Codes

The Berlekamp Welch Decoder

Inputs:

- ▶ Degree bound $k - 1$, and error bound $e = \left\lfloor \frac{q-k+1}{2} \right\rfloor$.
 - ▶ Corrupted evaluations $\{x_i, y_i\}_{i=1}^n$.
1. Calculate a polynomial $P(X)$ of degree e and a $Q(X)$ of degree $e + k - 1$ such that

$$y_i P(x_i) = Q(x_i) \quad \text{for } 1 \leq i \leq q$$

Notice: q linear equations

$(e + 1) + (e + k) = 2e + k + 1 = q + 2$ unknowns

2. Output

$$f(X) = \frac{Q(X)}{P(X)}$$

Reed Muller Codes

- ▶ A message will be a *multivariate* polynomial, $f \in \mathbb{F}_q[z_1, \dots, z_m]$ of total degree t .
- ▶ A codeword will be the evaluation of f at points in \mathbb{F}_q^m .
- ▶ The messages are then $\binom{m+t}{m}$ symbols in \mathbb{F}_q .
- ▶ Codewords are of length q^m .
- ▶ The Schwartz-Zippel Lemma tells us that

$$\Pr_{z_1, \dots, z_m \leftarrow S} [f(z_1, \dots, z_m) = 0] \leq \frac{t}{|S|}.$$

- ▶ The minimum distance is $\frac{t}{q}q^m = tq^{m-1}$.
- ▶ Reed-Muller codes can have small alphabet size

The Hadamard Code

A Binary Reed-Muller Code of Degree 1

Hadamard Code

- ▶ A message will be a degree 1 polynomial in $f \in \mathbb{F}_2[X_1, \dots, X_k]$
- ▶ A codeword will be the sequence $f(x)$ for all $x \in \mathbb{F}_2^k$

Alternate Formulation

- ▶ A message will be a vector x in \mathbb{F}_2^k
- ▶ A codeword will be the sequence $\langle x, r \rangle$ for all $r \in F_2^k$

Properties

- ▶ The rate is $\frac{k}{2^k} \rightarrow 0$
- ▶ The relative distance is $\frac{1}{2}$

Outline

Error Correcting Codes

List Decodable Codes

Locally-Decodable Codes

Matching Vector Codes

Multiplicity Codes

List Decodable Codes

Unique Decoding

A code is decodable up to rate ρ if every ball of radius ρn contains at most 1 codeword.

List Decoding

A code C is (ρ, L) -list decodable if every ball of radius ρn contains at most L codewords.

Benefits of List Decoding

- ▶ Singleton bound says unique decoding radius is at most $\frac{1}{2}(1 - R) \leq \frac{1}{2}$.
- ▶ Can we decode beyond error rate $\frac{1}{2}$?
- ▶ Applications to complexity:
 - ▶ Constructing hard-core predicates
 - ▶ Constructing randomness extractors

List Decoding Capacity

Theorem

List Decoding Capacity Let $q \geq 2$, and $0 \leq \rho \leq 1 - \frac{1}{q}$

1. For any L there exists a (ρ, L) -List decodable code with rate

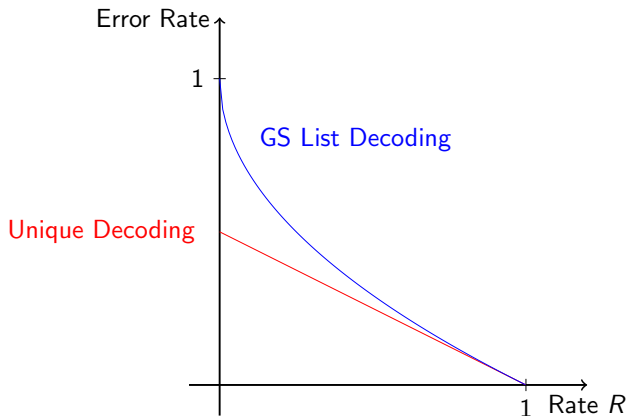
$$R \leq 1 - H_q(\rho) - \frac{1}{L}$$

2. For every (ρ, L) code of rate $R \geq 1 - H_q(\rho) + \epsilon$, L must be exponential in the block length of the code.

Remark: point 1 even holds for random linear codes with high probability [GHK10].

List-Decoding Reed-Solomon [Sud97, GS99]

- ▶ RS codes can be uniquely decoded up to error fraction $\frac{1}{2}(1 - R)$
- ▶ GS show how to list decode RS codes up to a distance $1 - \sqrt{R}$



Folded Reed Solomon Codes [GR06]

Reed Solomon Codes

$$f \mapsto (f(\alpha_1), \dots, f(\alpha_n))$$

Folded Reed Solomon Codes

$$f \mapsto \left(\begin{bmatrix} f(\alpha_1) \\ \vdots \\ f(\alpha_m) \end{bmatrix}, \begin{bmatrix} f(\alpha_{m+1}) \\ \vdots \\ f(\alpha_{2m}) \end{bmatrix}, \dots, \begin{bmatrix} f(\alpha_{n-m+1}) \\ \vdots \\ f(\alpha_n) \end{bmatrix} \right)$$

Folded Reed Solomon Codes

Performance of Folded Reed Solomon Codes

For any s with $1 \leq s \leq m$,

- ▶ Error rate $\approx 1 - \left(\frac{mR}{m-s+1}\right)^{\frac{s}{s+1}}$.
- ▶ List Size $\approx q^{s-1}$.
- ▶ Decoding time $\approx q^{s-1}$

- ▶ When $s = 1$ the code tolerates error rate $1 - \sqrt{R}$ which matches the Guruswami-Sudan list decoder for traditional Reed Solomon codes.
- ▶ Picking $s = 1/\epsilon$ and $m = 1/\epsilon^2$, the code tolerates an error rate greater than $1 - R - \epsilon$.
- ▶ With these parameters list size grows exponentially in $1/\epsilon$.

Derivative Codes [GW11]

Folded Reed Solomon Codes

$$f \mapsto \left(\left[\begin{array}{c} f(\alpha_1) \\ \vdots \\ f(\alpha_m) \end{array} \right], \left[\begin{array}{c} f(\alpha_{m+1}) \\ \vdots \\ f(\alpha_{2m}) \end{array} \right], \dots, \left[\begin{array}{c} f(\alpha_{n-m+1}) \\ \vdots \\ f(\alpha_n) \end{array} \right] \right)$$

Derivative Codes

$$f \mapsto \left(\left[\begin{array}{c} f(\alpha_1) \\ f'(\alpha_1) \\ \vdots \\ f^{(m-1)}(\alpha_1) \end{array} \right], \left[\begin{array}{c} f(\alpha_2) \\ f'(\alpha_2) \\ \vdots \\ f^{(m-1)}(\alpha_2) \end{array} \right], \dots, \left[\begin{array}{c} f(\alpha_n) \\ f'(\alpha_n) \\ \vdots \\ f^{(m-1)}(\alpha_n) \end{array} \right] \right)$$

Derivative Codes

Performance of Derivative Codes

For any s with $1 \leq s \leq m$,

- ▶ Error rate $\approx \frac{s}{s+1} \left(1 - \frac{mR}{m-s+1} \right)$.
- ▶ List Size $\approx q^{s-1}$.
- ▶ Picking $s = 1/\epsilon$ and $m = 1/\epsilon^2$, the code tolerates an error rate greater than $1 - R - \epsilon$.
- ▶ With these parameters list size grows exponentially in $1/\epsilon$.
- ▶ Random codes are (inefficiently) list decodable with lists of size $O(1/\epsilon)$.

List Decoding of the Hadamard Code [GL89]

Given oracle access to a (corrupted) codeword $\{\langle x, r \rangle\}_{r \in \{0,1\}^k}$

1. Select $s^1, \dots, s^\ell \stackrel{\$}{\leftarrow} \{0, 1\}^k$
2. Set $r^J = \bigoplus_{j \in J} s^j \in \{0, 1\}^k$ for $J \subset [\ell]$
(The r^J are pairwise independent)
3. Guess the values $\sigma^j = \langle x, s^j \rangle$ for $j \in [\ell]$
(We will iterate through all 2^ℓ guesses)
4. Set $\rho^J = \bigoplus_{j \in J} \sigma^j \in \{0, 1\}$ for $J \subset [\ell]$
5. For each index i from 1 to k
 - ▶ Query $c^J = \langle x, r^J + e^i \rangle$ (correct with probability $1 - \delta$)
 - ▶ Set z_i to be the majority vote of $c^J \oplus \rho^J$ over all J .
(Notice: $x_i = \langle x, r^J \oplus e^i \rangle \oplus \langle x, r^J \rangle$)

Add $z = z_1 \cdots z_k$ to the list

6. Return to step 3.
7. Output all 2^ℓ values of z .

List Decoding of the Hadamard Code

- ▶ Assume all ℓ guesses are correct, i.e. $\sigma^j = \langle x, s^j \rangle$.
- ▶ The vectors $r^j + e^i$ are uniformly distributed and pairwise independent.
- ▶ For a fixed i , each query $\langle x, r^j + e^i \rangle$ is correct with probability $1 - \delta$ and they are pairwise independent.
- ▶ If the error rate $\delta = \frac{1}{2} - \epsilon$, then we expect $(\frac{1}{2} + \epsilon)2^\ell$ queries are correct, and Chebyshev tells us that the probability that the majority is incorrect is bounded by $\frac{1}{\epsilon^2 2^\ell}$.
- ▶ Since we have to sum over all indices $i = 1, \dots, n$ setting $2^\ell = O(k/\epsilon^2)$ allows us to take a union bound over all indices and achieve constant probability of success.

Hard-Core Predicates from List-Decoding [AGS03]

1. Define a Code:

Given a predictor P define a code C^P such that given $f(x)$ and the ability to compute $P(z)$ from $f(z)$ yields query access to the codeword $C^P(x)$.

2. List Decode:

Find an efficient list-decoding algorithm for C^P that can tolerate an error rate of $\frac{1}{2} - \epsilon$.

3. Access a corrupted codeword:

Show that an adversary that can predict $P(x)$ from $f(x)$ can be used to simulate query access to a corrupted codeword of x with at most a $\frac{1}{2} - \epsilon$ fraction of errors.

Outline

Error Correcting Codes

List Decodable Codes

Locally-Decodable Codes

 Matching Vector Codes

 Multiplicity Codes

Locally Decodable Codes

Definition [KT00, STV99]

Definition

A code $C : \mathbb{F}_q^k \rightarrow \mathbb{F}_q^N$ is called (r, δ, ϵ) -locally decodable if

1. For all $x \in \mathbb{F}_q^k$, $i \in [k]$

$$\Pr[A^y(i) = x_i] \geq 1 - \epsilon$$

whenever $d(C(x), y) \leq \delta N$.

2. The decoding algorithm, A , makes at most r queries to y .

Locally Decoding the Hadamard Code

Given oracle access to a (corrupted) codeword

$$\{\langle x, r \rangle\}_{r \in \{0,1\}^k}$$

- ▶ To recover the i th bit, x_i , select r at random and query $\langle x, r \rangle$ and $\langle x, r \oplus e^i \rangle$
- ▶ Calculate $x_i = \langle x, e^i \rangle = \langle x, r \rangle \oplus \langle x, r \oplus e^i \rangle$
- ▶ The probability a uniformly chosen location of $C(x)$ is corrupted is $1 - \delta$
- ▶ By a union bound both queried locations are uncorrupted with probability at least $1 - 2\delta$
- ▶ Rate is $k/2^k \rightarrow 0$
- ▶ Locality is 2

Locally Decodable Codes

Different Regimes

Low Rate and Small Locality	Matching Vector Codes	[Yek07, Efr09]
Moderate Rate and Locality	Reed Muller Codes	
High Rate and Large Locality	Multiplicity Codes	[KSY11]

Locally Correcting Reed Muller Codes

- ▶ Given a message $f \in \mathbb{F}_q[z_1, \dots, z_m]$, and an index $w \in \mathbb{F}_q^m$
- ▶ To recover $f(w)$, the decoder picks a random affine line through w , i.e. it picks a random $v \in \mathbb{F}_q^m$, and the decoder reads

$$f(w + \lambda v)$$

for $\lambda \in \mathbb{F}_q$.

- ▶ The restriction of f to this line is a univariate polynomial of degree t . The value $f(w)$ can then be recovered by decoding a Reed-Solomon code (e.g. using Berlekamp-Welch).

Locally Correcting Reed Muller Codes

- ▶ Each query into the codeword is uniformly (but not independently) distributed.
- ▶ Each query hits a corrupt location with probability $1 - \delta$.
- ▶ We expect $(1 - \delta)(q - 1)$ of the queried locations are corrupted.
- ▶ Suppose the implied Reed Solomon code has minimum distance $(1 - \sigma)$, so it can decode from a $(1 - \sigma)/2$ fraction of errors.
- ▶ Markov's inequality tells us that the probability that more than $(1 - \sigma)(q - 1)/2$ locations are corrupted is bounded by $2\delta/(1 - \sigma)$.
- ▶ This can be adapted to decoding along a random degree two curve [GS92], now errors among the points are pairwise independent so we can use Chebyshev instead of Markov.

Parameters for Reed Muller Local Decoding

- ▶ A message is a multivariate polynomial, $f \in \mathbb{F}_q[z_1, \dots, z_m]$ of total degree t .
- ▶ **Block Length:** $n = q^m$
- ▶ **Rate:** $R = \frac{\binom{m+t}{m}}{q^m}$
- ▶ **Relative Distance:** t/q
- ▶ **Locality:** $q = \sqrt[m]{n}$

Outline

Error Correcting Codes

List Decodable Codes

Locally-Decodable Codes

 Matching Vector Codes

 Multiplicity Codes

Linear Algebra in the Exponent

Definition

Define $a \star b = a^b$, and extend naturally to matrix arithmetic. This operation is not commutative or associative.

Examples:



$$(a_1, \dots, a_n) \star \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} = \prod_{i=1}^n a_i^{b_i}$$



$$(a_1) \star (b_1, \dots, b_n) = (a_1^{b_1}, \dots, a_1^{b_n})$$



$$\begin{aligned} ((g_1) \star (w_1, \dots, w_n)) \star \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} &= (g_1^{w_1}, \dots, g_1^{w_n}) \star \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} \\ &= \prod_{i=1}^n g_1^{w_i u_i} = g_1^{u \cdot w} \end{aligned}$$

Matching Subsets

Definition

Let $S \subset \mathbb{Z}_n \setminus \{0\}$. Sets $U = \{u_1, \dots, u_k\} \subset \mathbb{Z}_n^m$ and $V = \{v_1, \dots, v_k\} \subset \mathbb{Z}_n^m$ form an S -matching if

1. $u_i \cdot v_i = 0$ for all $i \in [m]$
2. $u_i \cdot v_j \in S$ for all $i \neq j \in [m]$

Goal: maximize k

Encoding Using Matching Vector Codes

- ▶ Fix an S -matching $\{u_i\}, \{v_i\}$
- ▶ In the ring $\mathbb{F}_q[z_1, \dots, z_m]$, $z \star u = \prod_{i=1}^m z_i^{u_i}$ is a monomial of degree $\sum_i u_i$
- ▶ A message $x \in \mathbb{F}_q^k$ is first associated to a polynomial $f_x \in \mathbb{F}_q[z_1, \dots, z_m]$ as a sum of monomials

$$x \mapsto \sum_{j=1}^k x_j (z \star u_j)$$

- ▶ Encoding is done by evaluating f at all points in \mathbb{G}^m , where $\mathbb{G} = \langle g \rangle$

Decoding

Noiseless Setting

- ▶ To recover the i th symbol x_i
- ▶ Pick a random $w \in \mathbb{F}_q^m$ and construct the multiplicative line $\{g \star (w + \lambda v_i)\}$.
- ▶ Along this line

$$f(g \star (w + \lambda v_i)) = \sum_{j=1}^k x_j (g \star (w + \lambda v_i)) \star u_j = \sum_{j=1}^k x_j g^{u_j \cdot w} (g^\lambda)^{u_j \cdot v_i}$$

- ▶ By the properties of the matching $u_i \cdot v_i = 0$, and $u_j \cdot v_i \in S$

$$f(g \star (w + \lambda v_i)) = x_i g^{u_i \cdot w} + \sum_{s \in S} \left(\sum_{j: u_j \cdot v_i = s} x_j g^{u_j \cdot w} \right) g^{\lambda s}$$

- ▶ Consider the polynomial

$$p(y) = x_i g^{u_i \cdot w} + \sum_{s \in S} \left(\sum_{j: u_j \cdot v_i = s} x_j g^{u_j \cdot w} \right) y^s$$

- ▶ We know the evaluation of $p(y)$ at q points and we wish to find $x_i = p(0)/g^{u_i \cdot w}$. This can be done via Reed-Solomon decoding.

Performance of Matching Vector Codes

Performance is determined by size of S -matching

	Matching Vector Codes
Msg Length	k
Codeword Length	$e^{e^{(\log k)^{1/t}(\log \log k)^{1-1/t}}}$
Locality	$r = 3 \cdot 2^{t-2}$
Error Tolerance	$O(1/r)$

There are many variants. See [Yek10] for a survey.

Outline

Error Correcting Codes

List Decodable Codes

Locally-Decodable Codes

Matching Vector Codes

Multiplicity Codes

Multiplicity Codes [KSY11]

Reed Muller Codes (multivariate)

$$f \mapsto (f(\alpha_1), \dots, f(\alpha_n))$$

Multiplicity Codes [KSY11]

Reed Muller Codes (multivariate)

$$f \mapsto (f(\alpha_1), \dots, f(\alpha_n))$$

Bivariate Multiplicity Codes

$$f \mapsto \left(\left[\begin{array}{c} f(\alpha_1) \\ \frac{\partial f}{\partial X}(\alpha_1) \\ \frac{\partial f}{\partial Y}(\alpha_1) \end{array} \right], \dots, \left[\begin{array}{c} f(\alpha_n) \\ \frac{\partial f}{\partial X}(\alpha_n) \\ \frac{\partial f}{\partial Y}(\alpha_n) \end{array} \right] \right)$$

Multiplicity Codes [KSY11]

Reed Muller Codes (multivariate)

$$f \mapsto (f(\alpha_1), \dots, f(\alpha_n))$$

Bivariate Multiplicity Codes

$$f \mapsto \left(\begin{bmatrix} f(\alpha_1) \\ \frac{\partial f}{\partial X}(\alpha_1) \\ \frac{\partial f}{\partial Y}(\alpha_1) \end{bmatrix}, \dots, \begin{bmatrix} f(\alpha_n) \\ \frac{\partial f}{\partial X}(\alpha_n) \\ \frac{\partial f}{\partial Y}(\alpha_n) \end{bmatrix} \right)$$

Derivative Codes of Order 2 (univariate, for list decoding)

$$f \mapsto \left(\begin{bmatrix} f(\alpha_1) \\ f'(\alpha_1) \end{bmatrix}, \dots, \begin{bmatrix} f(\alpha_n) \\ f'(\alpha_n) \end{bmatrix} \right)$$

Decoding Bivariate Multiplicity Codes

► **Input:**

$$\text{A codeword } C(f) = \left(\begin{bmatrix} f(\alpha) \\ \frac{\partial f}{\partial X}(\alpha) \\ \frac{\partial f}{\partial Y}(\alpha) \end{bmatrix} \right)_{\alpha \in \mathbb{F}_q^2}$$

An index $a \in \mathbb{F}_q^2$ (for local correction).

► **Decoding:**

1. Pick $b \in \mathbb{F}_q^2$ and read the coordinates along the line $\{a + \lambda b\}_{\lambda \in \mathbb{F}_q}$.
2. We retrieve a univariate polynomial $g(\lambda) = f(a + b\lambda)$.
Observe: $g'(\lambda) = b_1 \frac{\partial f}{\partial X}(a + b\lambda) + b_2 \frac{\partial f}{\partial Y}(a + b\lambda)$, so we can recover $g'(\lambda)$ as well.
3. Notice that $g(0) = f(a)$ and $g'(0) = b_1 \frac{\partial f}{\partial X}(a) + b_2 \frac{\partial f}{\partial Y}(a)$.
We want to recover $f(a)$, $\frac{\partial f}{\partial X}(a)$, $\frac{\partial f}{\partial Y}(a)$.
4. Repeat step 2 for different choice of b , and solve the 2 linear equations for $\frac{\partial f}{\partial X}(a)$ and $\frac{\partial f}{\partial Y}(a)$.

Performance of Bivariate Multiplicity Codes

Fix a relative distance δ and field \mathbb{F}_q

	Reed Muller	Bivariate Multiplicity
Degree	$t = (1 - \delta)q$	$t = 2(1 - \delta)/q$
Msg Length	$k = \binom{t+1}{2}$	$k = \binom{t+1}{2}$
Codeword Length	$n = q^2$	$n = 3q^2$
Rate	$\frac{k}{q^2} \approx \frac{t^2}{2q^2} = (1 - \delta)^2/2 < \frac{1}{2}$	$\frac{k}{3q^2} \approx \frac{t^2}{6q^2} = 2(1 - \delta)^2/3$
Locality	$q = \sqrt{n}$	$2q$

Multiplicity Codes

Performance of Multiplicity Codes

For any rate R , and every $\epsilon > 0$, there are multiplicity codes with

- ▶ **Relative Distance:** $\delta = \epsilon(1 - R)/2$
(Recall the singleton bound says $\delta \leq (1 - R)$)
- ▶ **Error Tolerance:** $\epsilon(1 - R)/20$
- ▶ **Locality:** $O(n^\epsilon)$

Thanks!



Adi Akavia, Shafi Goldwasser, and Muli Safra.

A unifying approach for proving hardcore predicates using list decoding.

In *FOCS '03*, 2003.



Emmanuel Candes, Mark Rudelson, Terence Tao, and Roman Vershynin.

Error correction via linear programming.

In *FOCS '05*, volume 46, pages 668 – 681, 2005.



Alexandros G. Dimakis and Pascal O. Vontobel.

Lp decoding meets lp decoding: A connection between channel coding and compressed sensing.






In *Allerton*, 2009.



Klim Efremenko.

3-query locally decodable codes of subexponential length.

In *STOC '09*, pages 39–44. ACM, 2009.

-  Venkatesan Guruswami, Johan Håstad, and Swastik Kopparty.
On the list-decodability of random linear codes.
In *STOC '10*, pages 409–416, 2010.
-  Oded Goldreich and Leonid Levin.
A hard-core predicate for all one-way functions.
In *STOC '89*, 1989.
-  Parikshit Gopalan, Richard J. Lipton, and Yan Z. Ding.
Error correction against computationally bounded adversaries.
Manuscript, 2004.
-  Venkatesan Guruswami and Atri Rudra.
Explicit capacity-achieving list-decodable codes.
In *STOC '06*, 2006.
-  Peter Gemmel and Madhu Sudan.
Highly resilient correctors for polynomials.
Information Processing Letters, 43:169–174, 1992.



Venkatesan Guruswami and Madhu Sudan.

Improved decoding of reed-solomon and algebraic-geometric codes.

Transactions on Information Theory, 45:1757–1767, 1999.



Venkatesan Guruswami and Adam Smith.

Codes for computationally simple channels: Explicit constructions with optimal rate.

In *FOCS '10*, 2010.



Venkatesan Guruswami and Carol Wang.

Optimal rate list decoding via derivative codes.





In *RANDOM '11*, 2011.



Brett Hemenway and Rafail Ostrovsky.

Public-key locally-decodable codes.

In *CRYPTO*, pages 126–143, 2008.

-  Brett Hemenway, Rafail Ostrovsky, Martin Strauss, and Mary Wootters.
Public key locally decodable codes with short keys.
In *RANDOM '11*, 2011.
-  Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin.
High-rate codes with sublinear-time decoding.
In *STOC '11*, 2011.
-  Jonathan Katz and Luca Trevisan.
On the efficiency of local decoding procedures for error-correcting codes.
In *STOC '00: Proceedings of the 32nd Annual Symposium on the Theory of Computing*, pages 80–86, 2000.
-  Richard J. Lipton.
A new approach to information theory.
In *STACS '94: Proceedings of the 11th Annual Symposium on Theoretical Aspects of Computer Science*, pages 699–708, London, UK, 1994. Springer-Verlag.



Silvio Micali, Chris Peikert, Madhu Sudan, and David A. Wilson.

Optimal error correction against computationally bounded noise.

In Joe Kilian, editor, *TCC*, volume 3378 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2005.



Rafail Ostrovsky, Omkant Pandey, and Amit Sahai.





Private locally decodable codes.

In *ICALP '07 : Proceedings of the 34th International Colloquium on Automata, Languages and Programming*, volume 4596 of *Lecture Notes in Computer Science*, pages 387–298. Springer, 2007.



Atri Rudra.

List Decoding and Property Testing of Error Correcting Codes.
PhD thesis, University of Washington, 2007.

-  Madhu Sudan, Luca Trevisan, and Salil Vadhan.
Pseudorandom generators without the xor lemma.
In *STOC '99*, 1999.
-  Madhu Sudan.
Decoding of reed-solomon codes beyond the error-correction bound.
Journal of Complexity, 13(1):180–193, 1997.
-  Sergey Yekhanin.
Towards 3-query locally decodable codes of subexponential length.
In *STOC '08*, pages 266–274. ACM, 2007.
-  Sergey Yekhanin.
Locally decodable codes.
Foundations and Trends in Theoretical Computer Science, 2010.



Sergey Yekhanin.

Locally decodable codes: A brief survey.

In *IWCC '11*, 2011.