Reed-Muller Error-Correcting Codes

Prateek Sharma

April 30, 2010

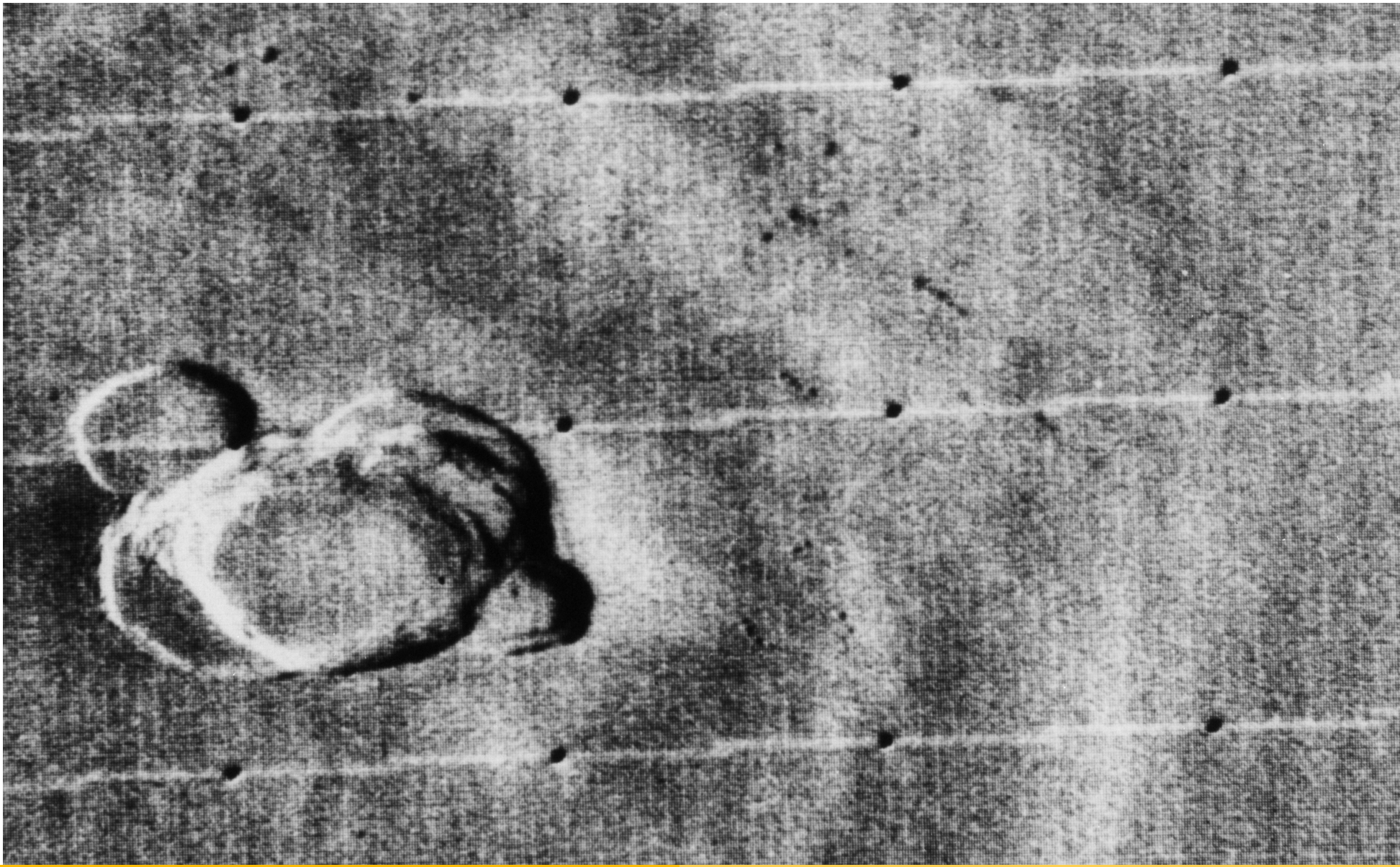
Reed-Muller codes

Second oldest codes (1954) after Hamming and the Golay codes. In this talk:

- Code properties: distance, dimension, orthogonal code
- Decoding Algorithms
- Applications of Reed-Muller codes

Used in the Mariner-9 spacecraft in 1972
Fast decoding algorithms (the Green machine).

Mariner-9



Coding theory concepts

Linear Code of length n and rank k :
A linear subspace C with dimension k of the vector space $\mathbb{F}_q^n$ where $\mathbb{F}_q$ is the finite field with q elements.
Hamming Distance: Number of differing positions
Distance: The minimum distance between all pairs of codewords
Parity Check: $\mathbf{x} = 01110011.\ x_6 = x_0 + x_1$ $\mathbf{x}' = 11110011.\ x_6 = x_0 + x_1$

Construction

$$\mathbf{x} \equiv (x_1, x_2, \ldots, x_m) \in \mathbb{F}^m$$
$$f(\mathbf{x}) \quad f : \mathbb{F}_2^m \rightarrow \{0, 1\}$$

Truth-Table

x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
f	0	1	0	1	0	1	1	0

f is a $n = 2^m$ length vector over F_2
Disjunctive Normal Form : $f = x_3 + x_1x_2$ ($+$ is *xor*)
A collection of 2^{2^m} vectors, each of length 2^m.

Boolean Monomials

$$M = \{1, x_1, x_2, \ldots, x_m, x_1x_2, \ldots, x_{m-1}x_m, x_1x_2x_3, \ldots, x_1x_2 \ldots x_m\}$$

$$f = 1 + a_1x_1 + a_2x_2 + \ldots + a_mx_m + a_{12}x_1x_2 + \ldots + a_{12\ldots r}x_1x_2 \ldots x_r + \ldots$$

Since **f** is a linear combination, it follows that the length of $x_1, x_2, \ldots, x_m$ is 2^m.

Reed-Muller Codes

Reed-Muller codes

The Reed-Muller codes of order r and length $n = 2^m$,$0 \leq r \leq m$ is the set of all vectors $\mathbf{f}$, where $f(x_1, \ldots, x_m)$ is a Boolean function which is a polynomial of degree at most r.

First-order codes

$\mathbf{1} + a_1 x_1 + a_2 x_2 + \ldots + a_m x_m$

Linearity

Lemma

$\mathcal{R}(r, m)$ is a linear code.

Basis

The monomials of degree $\leq r$ form a basis for $\mathcal{R}(r, m)$.

Generator matrix

$$G(r,m) = \begin{pmatrix} \mathbf{1} \\ \mathbf{x_1} \\ \mathbf{x_2} \\ \vdots \\ \mathbf{x_m} \\ \mathbf{x_1 x_2} \\ \vdots \\ \mathbf{x_1 x_2 \ldots x_r} \end{pmatrix}$$

Dimension

The dimension (k) of $\mathcal{R}(r, m)$ is equal to the number of monomials of degree $\leq r$ $\quad k = 1 + \binom{m}{1} + \binom{m}{2} + \ldots + \binom{m}{r}$

$\mathcal{R}(0, m)$ is the repetition code (2^m repetition).
$\mathcal{R}(m, m)$ consists of all possible binary sequences of length 2^m.

Length	$n = 2^m$
Minimum Distance	$d = 2^{m-r}$
Dimension	$k = 1 + \binom{m}{1} + \binom{m}{2} + \ldots + \binom{m}{r}$

Example code

$\mathcal{R}(1,3) =$

1	1	1	1	1	1	1	1	1
x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
$x_1 + x_2$	0	0	1	1	1	1	0	0
$x_1 + x_3$	0	1	1	0	0	1	1	0
$x_2 + x_3$	0	1	0	1	1	0	1	0
$x_1 + x_2 + x_3$	0	1	1	0	1	0	0	1
$1 + x_1$	1	1	1	1	0	0	0	0
$1 + x_2$	1	1	0	0	1	1	0	0
$1 + x_3$	1	0	1	0	1	0	1	0
$1 + x_1 + x_2$	1	1	0	0	0	0	1	1
$1 + x_1 + x_3$	1	0	0	1	1	0	0	1
$1 + x_2 + x_3$	1	0	1	0	0	1	0	1
$1 + x_1 + x_2 + x_3$	1	0	0	1	0	1	1	0

$$G(2,3) = \begin{array}{|l|cccccccc|} \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x_1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ x_2 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ x_3 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ x_1 \cdot x_2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ x_1 \cdot x_3 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ x_2 \cdot x_3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hline \end{array}$$

Recursive Definition

Theorem

$$\mathcal{R}(r+1, m+1) = \{\mathbf{u}|\mathbf{u}+\mathbf{v} : \mathbf{u} \in \mathcal{R}(r+1, m), \mathbf{v} \in \mathcal{R}(r, m)\}$$

$$G(r+1, m+1) = \begin{pmatrix} G(r+1, m) & G(r+1, m) \\ 0 & G(r, m) \end{pmatrix}$$

$$G(1, m+1) = \begin{pmatrix} G(1, m) & G(1, m) \\ 0 & 1 \end{pmatrix}$$

where

$$G(0, m) = (\overbrace{\mathbf{1111}}^{2^m})$$

Properties

Nested Structure

$$\mathcal{R}(r,m) \subseteq \mathcal{R}(t,m) \quad \text{if } 0 \leq r \leq t \leq m$$

Concatenated codes

Let C_i be an $[n, k_i, d_i]$ code. Then the concatenated code defined by

$$C = \{(\mathbf{u}, \mathbf{u}+\mathbf{v}) | \mathbf{u} \in C_1, \mathbf{v} \in C_2\}$$

has the parameters

$$[2n, k_1 + k_2, min\{2d_1, d_2\}]$$

Properties

Distance

Minimum distance, $d = 2^{m-r}$

First-Order

Every codeword in $\mathcal{R}(1, m)$ (except $\mathbf{1}, \mathbf{0}$) has weight 2^{m-1}

Dimension

$$dim(\mathcal{R}(r, m)) = dim(\mathcal{R}(r, m-1)) + dim(\mathcal{R}(r-1, m-1))$$

Uniqueness

Any linear code with parameters $[2^m, m+1, 2^{m-1}]$ is equivalent to the first order Reed-Muller code.

Dual and Orthogonal of RM codes

Dual

$\mathcal{R}(m-r-1,m) = \mathcal{R}(r,m)^{\perp}$

The dual code $\mathcal{R}(1,m)^{\perp}$ is the extended binary Hamming code $H(m)$
The *Orthogonal code* $\mathcal{O}_m$ is a $[2^m, m, 2^{m?1}]$ code consisting of the vectors $\sum_{i=1}^{m} u_i \mathbf{v_i}$

Orthogonal Code

$\mathcal{R}(1,m) = \mathcal{O}_m \cup (\mathbf{1} + \mathcal{O}_m)$

Plotkin Bound

Plotkin Bound

If $C = [n, k, d]$ code,

$$d \leq \frac{n2^{k-1}}{2^k - 1}$$

Proof.

Counting in two ways: □

Step Decoding

Theorem

One-step majority logic decoding can correct upto $\frac{n-1}{2(d'-1)}$ *errors.*
(d' is the minimum distance of the dual code.)

Proof.

$$\begin{array}{lll} & \overbrace{\quad}^{\geq d'-1} & \\ 1 & 111 & 00000 \\ 1 & 000 & \underbrace{11111}_{\geq d'-1} \end{array}$$

□

L-Step Decoding

$E_L \leq \frac{n}{d'} - \frac{1}{2}$

Reed Decoding Algorithm

Algorithm

1. For each row, find the 2^{m-r} characteristic vectors, and take the dot product with $\mathbf{x}$.
2. The majority of the values of the dot products is the coefficient of the row (0/1) .
3. Coefficient vector is the original message.

Example

Let $\mathbf{m} = 0110$ in $\mathcal{R}(1,3)$. $\mathbf{c} = 00111100$. $\mathbf{x} = 10111100$.

Hadamard Transform

The Real Vector $F(\mathbf{v})$ of a binary vector $\mathbf{v}$ is a vector with 0 replaced by 1 and 1 by -1.

Theorem

The Orthogonal code $\mathcal{O}_m$ with real vectors is equivalent to the Hadamard matrix H_m.

Proof.

$$v_i, v_j \in \mathcal{R}(1,m) \implies v_i \cdot v_j = 0$$

$\mathcal{O}_m$ is a $2^m x 2^m$ matrix with elements $+1, -1$ such that dot product of any two rows is 0. □

$$\mathcal{R}(1,m) = \begin{pmatrix} H_m \\ -H_m \end{pmatrix}$$

Maximize the correlation between received vector u and the rows:
$corr(F(u), F(v)) = n - d(u,v)$
$Max(v \cdot H_m)$ gives the row with the maximum correlation, and is the

Fast Hadamard Transform

Recursively calculate the correlation. Kronecker-product construction of the Hadamard Matrix.

$$H_{2^m} = M_{2^m}^{(1)} M_{2^m}^{(2)} \dots M_{2^m}^{(m)}$$

where

$$M_{2^m}^{(i)} = I_{2^{m-i}} \otimes H_2 \otimes I_{2^{i-1}}, \qquad 1 \leq i \leq m$$

Easy to implement in hardware — used in Mariner-9. (Green Machine) $O(n \log n)$ time. Fastest 'classical' decoding algorithm for the first-order codes.

List Decoding

List Decoding algorithm for $\mathcal{R}(1, m)$ capable of correcting $n(\frac{1}{2} - \epsilon)$ errors in $O(n\epsilon^{-3})$.

Hadamard Transform: $\frac{n}{4}$ errors in $O(n \log n)$ time.

The algorithm performs a 'smart' Hadamard Transform. Builds a list of codewords within some distance Δ of the received vector $\mathbf{y}$

List: $L_\epsilon(y) = \{f \in \mathcal{R}(1, m) : d(\mathbf{y}, \mathbf{f}) \leq n(\frac{1}{2} - \epsilon)\}$

Facet: $S_\alpha = \{(x_1, \ldots x_i, \alpha_{i+1} \ldots \alpha_m\} \quad \alpha = (\alpha_{i+1}, \ldots, \alpha_m)$

List Decoding Algorithm

Candidate (ith prefix): $c^{(i)}(x_1, \ldots, x_m) = c_1 x_1 + \ldots + c_i x_i$

iteration-i: $L_\epsilon^{(i)}(y) = \{c^{(i)}(x_1, x_2, \ldots, x_i) = c^{(i-1)} + c_i x_i\}$

$\mathbf{v}_\alpha^{(\mathbf{i})} \equiv \mathbf{y}_\alpha \mathbf{c}^{(\mathbf{i})}$ where $\mathbf{y}_\alpha$ is the restriction of y on the Boolean-facet α

Distance

$$\Delta(\mathbf{y}, \mathbf{c}^{(\mathbf{i})}) = \sum_\alpha |\mathbf{y}_\alpha \mathbf{c}^{(\mathbf{i})}| = \sum_\alpha |\mathbf{v}_\alpha^{(\mathbf{i})}|$$

$$\mathrm{L}_\epsilon^{(i)}(y) = \{c^{(i1)} : \Delta(\mathbf{y}, \mathbf{c}^{(\mathbf{i})}) \geq n\epsilon\}$$

List Decoding

Recursive facets

$S_{0,\alpha} = \{(x_1, \ldots x_{i-1}, 0, \alpha_{i+1} \ldots \alpha_m\}$
$S_{1,\alpha} = \{(x_1, \ldots x_{i-1}, 1, \alpha_{i+1} \ldots \alpha_m\}$

$c^{(i)} = c^{(i-1)} + c_i x_i : \mathbf{v}_{\alpha}^{(\mathbf{i})} = \mathbf{v}_{\mathbf{0},\alpha}^{(\mathbf{i})} + (-1)^{c_i} \mathbf{v}_{\mathbf{1},\alpha}^{(\mathbf{i-1})}$

Example

¡todo¿

Applications

Communication Used in Mariner and Viking space-probes in the 1970's. More recently, used is in the IEEE 802.11b standard for Wireless Local Area Networks (WLANs).

Testing Low-degree polynomials Using $\mathcal{R}(1, m)$ codes to test whether a binary function is a low-degree polynomial is a central theme in a lot of research in complexity theory [**?**] , [**?**].In a typical scenario, the Boolean functions are mapped to the Reed-Muller codes, and the properties are used to prove the bounds on the number of queries needed to determine the original function.

Sidelnikov cryptanalysis The cryptanalysis attack uses the properties of Reed-Muller codes to break the cryptographic code [**?**]. The uniqueness result stated earlier is a central feature in the cryptographic attack. [**?**]

Side Channel attacks The list decoding of the $\mathcal{R}(1, m)$ codes is used in the side-channel attack described in [**?**].

References

Future Work

- Second order codes. Have several interesting weight distribution properties
- Generalized Reed-Muller codes
- List Decoding for higher order codes

END

END