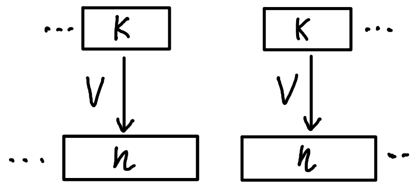
Classical linear codes

Applications of classical codes: wireless communication, DVDs, SSDs, RAM (for larger computers)

Block coding:

Logical data (messages):

Physical code blocks:



Encoding: $V: \{0,1\}^k \to \{0,1\}^n$

$C = \mathrm{Image}(V)$

Definition. A code $C \subseteq N$ *protects from a set of errors* $E \subseteq N \times N$ if

$$\forall x_1, x_2 \in C \quad \forall y_1, y_2 \text{ such that } (x_1, y_1), (x_2, y_2) \in E, \quad x_1 \neq x_2 \Rightarrow y_1 \neq y_2.$$

$$\boxed{\text{Code of type } [n,k]: \quad N = \{0,1\}^n, \quad |C| = 2^k.}$$

Error set (should include the most likely errors for a given application)

Independent bit flips

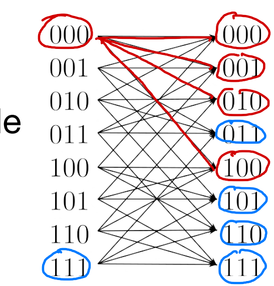
$$E = E(n,r) = \{ (x,y) : \ \mathrm{dist}(x,y) \leq r \}$$

Hamming distance (between binary words):

$$\boxed{\mathrm{dist}(x,y) = \#\text{ of distinct bits in } x, y}$$

Example:
The 3-bit repetition code protects from errors in

$E(3,1)$



Burst errors (e.g. scratches on a DVD): Allow r_1 consequtive and r_2 independent flips

Code distance: $\boxed{d = \min\{ dist(x_1, x_2) : \; x_1, x_2 \in C, \; x_1 \neq x_2 \}}$

Code of type *[n,k,d]*:

# of physical bits

# of logical bits

distance

Simplest examples

Repetition code:
(type *[n,1,n]*)

$$Rep(n) = \{0^n, 1^n\}$$

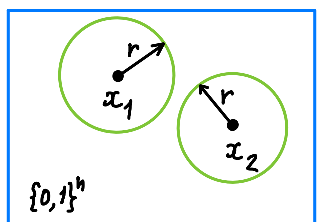
$$d = dist(0^n, 1^n) = n$$

Parity check:
(type *[n,n-1,2]*)

$$Par(n) = \{(u_1, \ldots, u_n) : \sum_j u_j \equiv 0 \mod 2\}$$

$$d = dist(0^n, 110\ldots0) = 2$$

Rep(n) protects from $\left\lfloor \frac{n-1}{2} \right\rfloor$ bit flips, *Par(n)* detects 1 flip



In general, code *C* $\begin{cases} \text{protects from } r \text{ errors if} & 2r < d \quad \Rightarrow \quad r_{max} = \left\lfloor \frac{d-1}{2} \right\rfloor \\ \text{detects } q \text{ errors if} & q < d \quad \Rightarrow \quad q_{max} = d-1 \end{cases}$

Linear codes

$C \subseteq \mathbb{F}_2^n$ is linear subspace of the *n*-dimensional vector space over the field $\mathbb{F}_2 = \{0,1\}$

(A field is a commutative ring in which every nonzero element is invertible)

(For example, the repetition and parity codes are linear)

$$dist(x-y) = |x-y|$$

(# of nonzero elements in x-y, a.k.a. the *Hamming weight*)

For linear codes, one can use the usual concepts of linear algebra: linear independence, basis, subspace dimensionality.

Generalization: Additive codes

N is an Abelian group, $C \subseteq N$ is a subgroup

Group of residues modulo *q*: $\mathbb{Z}_q = \{0,\dots,q-1\}$

$\mathbb{Z}_q = \mathbb{F}_q$ is a field if *q* is a prime number

In general, it is a ring (i.e. multiplication is defined)

$x \in \{0,\dots,q-1\}$ is an invertible element of $\mathbb{Z}_q$ if x and q are mutually prime

Examples

$$Rep_q(n) = \{(u,\dots,u):\ u \in \mathbb{Z}_q\}$$

$$Par_q(n) = \{(u_1,\dots,u_n):\ u_1,\dots,u_n \in \mathbb{Z}_q,\ \sum_j u_j \equiv 0 \mod q\}$$

Different descriptions of a linear code

By basis elements $g_1, .., g_k \in C$

(rows of the *generator matrix* G)

By linear equations, or a *check matrix* H

$$C = \{ u \in F^n : \ h_j u^T = 0 \ \text{ for } j = 1,..,n-k \}$$

The check matrix has rows $h_1, .., h_{n-k}$

Rep(5)

$$G_{Rep(5)} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$H_{Rep(5)} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Par(5)

$$G_{Par(5)} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

$$H_{Par(5)} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

The check matrix is the generator matrix for the *dual code*: $H_C = G_{C^\perp}$

$$\boxed{C^\perp = \{ v \in F^n : \ \forall u \in C \quad \underline{(v,u)} = 0 \}}$$

$$\boxed{(v,u) = v u^T = \sum_{j=1}^{n} v_j u_j \in F}$$

$$Rep(n)^\perp = Par(n), \qquad Par(n)^\perp = Rep(n)$$

In general, $(C^\perp)^\perp = C$

Caveat: Since the inner product is computed modulo 2, a vector can be orthogonal to itself

$$\begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 0, \qquad Rep(2)^\perp = Rep(2)$$

Hamming code $\mathrm{Ham}(m)$ of type $[2^m-1,\ 2^m-m-1]$

The 2^m-1 bits are indexed by nonzero binary numbers *x* of length *m*: $x = \overline{x_m \ldots x_1} = \sum_{s=1}^{m} x_s \cdot 2^{s-1}$

Rows of the check matrix: $(h_s)_x = H_{s,x} := x_s$ the s-th least significant bit of x

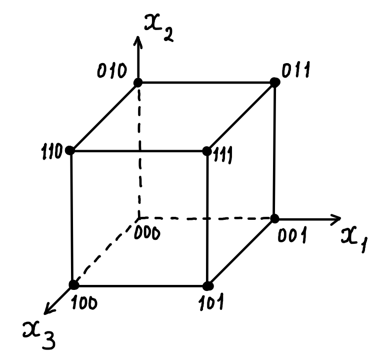
$u \in C \iff (h_s, u) := \sum_x x_s u_x = 0$ for $s=1,..,m$

Example: the 7-bit Hamming code (*m*=3)

$$u \in C \iff \begin{cases} u_{001} + u_{011} + u_{101} + u_{111} = 0 \\ u_{010} + u_{011} + u_{110} + u_{111} = 0 \\ u_{100} + u_{101} + u_{110} + u_{111} = 0 \end{cases}$$

$$H u^T = 0, \qquad H = \begin{array}{c} \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \\ \begin{array}{ccccccc} \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\ 001 & 010 & 011 & 100 & 101 & 110 & 111 \end{array} \end{array}$$

The 7 bits are associated with vertices of a 3-dimensional cube:



Description in terms of a basis or a generator matrix: $C = \text{lin. span.}\{g_1, g_2, g_3, g_4\}$

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

The Hamming code has distance 3 $\Longrightarrow$ the code protects from 1 bit flip

1) There is no $u \in Ham(m)$ with Hamming weight 1 or 2.

2) $\exists\ u \in Ham(m),\quad |u| = 3.$ (Easy: $u = (1,1,1,0,..,0)$)

Proof of (1): We will show that if $|u|=1$ or $|u|=2$, then $u \notin C$.

a) $|u| = 1:\quad u = (0\ldots 0\underbrace{1}_{x}0\ldots 0),$ i.e. $u_x = 1$ for a single $x = \overline{x_m \ldots x_1}$

$x_s = 1$ for some *s* $\Rightarrow (h_s, u) = x_s = 1 \Rightarrow u \notin C$

b) $|u| = 2:\quad u = (\cdots \underbrace{1}_{x} \cdots \underbrace{1}_{y} \cdots),$ i.e. $u_z = \begin{cases} 1, & \text{if } z=x,y \\ 0, & \text{otherwise} \end{cases}$

$x_s \neq y_s$ for some *s* $\Rightarrow (h_s, u) = x_s + y_s \neq 0 \Rightarrow u \notin C$

Extended Hamming code: add a bit x_{000} and the overall parity check: $h_0 = (1,..,1)$

type *[n,k,d], where* $n = 2^m,\quad K = 2^m - m - 1,\quad d = 4$

Reed-Muller codes $\boxed{RM(m, l) \text{ of type } [n,k], \text{ where } n = 2^m, \quad K = \sum_{p=0}^{l} \binom{m}{p}}$

We interpret binary words of length 2^m as functions $u: \{0,1\}^m \to \{0,1\}$

For example, $u=(0,1,1,1)$ is interpreted as the function

$x=(x_1,x_2)$	$u(x)$
$(0,0)$	0
$(0,1)$	1
$(1,0)$	1
$(1,1)$	1

$$u(x_1,x_2) = x_1 + x_2 + x_1 x_2$$

$x^{\{1\}}$ $x^{\{2\}}$ $x^{\{1,2\}}$

The codewords are multilinear polynomials in $x_1,\dots,x_m$ of degree $\leq l$:

$$u \in C \iff u(x) = \sum_{\substack{A \subseteq \{1,\dots,m\} \\ |A| \leq l}} c_A x^A$$ for some set of coefficients $c_A \in \{0,1\}$

where $x^A = \prod_{s \in A} x_s$ denotes a monomial with support *A*

Example: $u = 1 + x_2 + x_1 x_3 \in RM(3,2)$

Important special case: $RM(m,1) = \left\{ c_0 + \sum_{s=1}^{m} c_s x_s : c_0, c_1, \dots, c_m = 0,1 \right\} = (\text{Ext. Hamming } (m))^{\perp}$

Some properties of monomials

Monomials x^A form a basis of the space of functions $\{0,1\}^n \to \{0,1\}$

Part 1: the monomials span the space of functions

<u>Proof by induction in *m*</u>

The base case (*m=0*) is trivial

Induction step:
$$u(\underbrace{x_1,..,x_{m-1}}_{x'}, x_m) = \underbrace{u(x',0)}_{} + \underbrace{\left(u(x',1) - u(x',0)\right)}_{}\cdot x_m$$

sums of monomials in $x_1,..,x_{m-1}$

Part 2: the monomials are linearly independent

This follows from part 1 because # of monomials = 2^m = dimension of the space

For example, let *m=4*

$A=\{1,2\},\ B=\{2,3\}$

$x^A x^B = (x_1x_2)(x_2x_3) = x_1x_2x_3$

does not depend on x_4;

hence # of 1s is even

Inner product:
$$\boxed{(x^A, x^B) = \sum_x \underbrace{x^A x^B}_{x^{A\cup B}} = \begin{cases} 1 & \text{if } A\cup B = \{1,..,m\} \\ 0 & \text{otherwise} \end{cases}}$$

Corollary: $\underline{RM(m,\ell)^\perp}$ = lin. span $\{x^A : \underbrace{A\cup B \neq \{1,..,m\} \text{ for all } B \text{ s.t. } |B|\leq \ell}_{|A|<m-\ell}\} = \underline{RM(m, m-\ell-1)}$

The code RM(m, l) **has distance** $d = 2^{m-\ell}$

$d \leq 2^{m-\ell}$ because any monomial x^A such that $|A| = \ell$ has Hamming weight $2^{m-\ell}$

Proof that $d \geq 2^{m-\ell}$ by induction in m

Base case (m=0): $d\left(RM(0,0)\right) = 1$ because $RM(0,0)$ has type $[1,1]$ $\Rightarrow$ $C = \mathbb{F}_2 \subseteq \mathbb{F}_2$

Induction step: we assume that $d\left(RM(m-1, \ell')\right) \geq 2^{m-1-\ell'}$ for all ℓ'

Let $u \in RM(m, \ell)$, $u \neq 0$

Define $u_0, u_1 \in RM(m-1, \ell)$ as follows: $\boxed{u_a(x') := u(x', a), \quad a = 0, 1}$

$u_1 - u_0 \in RM(m-1, \ell-1)$ because $(x^A)_1 - (x^A)_0 = \begin{cases} 0 & \text{if } m \notin A \\ x^{A-\{m\}} & \text{if } m \in A \end{cases}$

Case 1: $u_0 = u_1 \neq 0$ $\Rightarrow$ $|u_0| \geq d\left(RM(m-1, \ell)\right) = 2^{m-1-\ell}$ $\Rightarrow |u| = 2|u_0| \geq 2^{m-\ell}$

Case 2: $u_1 - u_0 \neq 0$ $\Rightarrow$ $|u_1 - u_0| \geq d\left(RM(m-1, \ell-1)\right) = 2^{m-\ell}$ $\Rightarrow |u| \geq |u_1 - u_0| \geq 2^{m-\ell}$

Error correction algorithm

In general the error correction problem is NP-hard (may require exhaustive search through all codewords).

However, for Reed-Muller codes, there is an efficient algorithm.

$\boxed{\begin{array}{ll} \text{Input:} & \tilde{u} \in \{0,1\}^{2^m} \text{ such that } \exists\, u \in RM(m,\ell),\ |\tilde{u}-u| < 2^{m-\ell-1} \\ \text{Output:} & u \end{array}}$

Top-level procedure

let $v = \tilde{u}$

for $p = \ell, \dots, 0$

for all $A \subseteq \{1, \dots, m\}$ such that $|A| = p$

find C_A using v $\leftarrow$ main subroutine

update v as follows: $v(x) := v(x) - C_A x^A$

$$u = \sum_{A:\, |A| \le \ell} C_A x^A$$

Main subroutine

Input: $A \subseteq \{1, .., m\}, \quad |A| = \ell$

$\tilde{u} \in \{0,1\}^{2^m}$ such that $\exists\, u = \sum_{B : |B| \le \ell} C_B\, x^B, \quad |\tilde{u} - u| < 2^{m-\ell-1}$

Output: C_A

W.l.o.g. we may assume that $A = \{1, .., \ell\}$, $\quad x = (\underbrace{x_1, .., x_\ell}_{x_A}, \underbrace{x_{\ell+1}, .., x_m}_{x_{\bar{A}}})$

Consider these functions of $x_{\bar{A}}$:

$$f(x_{\bar{A}}) := \sum_{x_A} u(x_A, x_{\bar{A}}) = \sum_{x_A} \sum_B C_B \underbrace{(x)^B}_{=\, x_A^{B \cap A}\, x_{\bar{A}}^{B \cap \bar{A}}}$$

$$= \sum_B C_B \underbrace{\sum_{x_A} x_A^{B \cap A}}_{\substack{1 \text{ if } B \supseteq A \\ 0 \text{ otherwise}}} x_{\bar{A}}^{B \cap \bar{A}} = C_A$$

$$\tilde{f}(x_{\bar{A}}) := \sum_{x_A} \tilde{u}(x_A, x_{\bar{A}})$$

Bit string interpretation:

$f = (\underbrace{C_A}_{1 \text{ bit}}, ..., C_A)$ $\quad$ ($2^{m-\ell}$ bits)

$\tilde{f} = (\tilde{f}(\underbrace{...00}), \tilde{f}(\underbrace{...01}), \dots)$

$x_{\bar{A}}$ for different $\bar{A}$

Computing $C_A \in \{0,1\}$: $\quad |\tilde{f} - f| < 2^{m-\ell-1} \Rightarrow$ $\boxed{C_A = \mathrm{MAJ}\left(\tilde{f}(x_{\bar{A}}) : x_{\bar{A}} \in \{0,1\}^{2^{m-\ell}}\right)}$

Punctured Reed-Muller codes

Let us remove the bit with index $x = (0, \dots, 0)$.

The trivial monomial, $x^{\emptyset} = 1$ may or may not be included:

$$\boxed{\begin{array}{lll} RM'(m,\ell): & u(x) = \sum_{A:|A|\le \ell} c_A x^A & \left[2^m - 1,\ \sum_{p=0}^{\ell} \binom{m}{p},\ 2^{m-\ell} - 1\right] \\ RM''(m,\ell): & u(x) = \sum_{1 \le |A| \le \ell} c_A x^A & \left[2^m - 1,\ \sum_{p=1}^{\ell} \binom{m}{p},\ 2^{m-\ell}\right] \end{array}}$$

$$RM'(m,\ell)^{\perp} = RM''(m, m-\ell-1)$$

Special case: $\mathrm{Ham}(m) = \left(RM''(m,1)\right)^{\perp} = RM'(m, m-2)$