ERROR-CORRECTING CODE

$C \subseteq \Sigma^n$ $\quad \Sigma$ alphabet

$(n,k,d)_q$ - code

- block length n
- message length $k (= \log_q |C|)$
- minimal distance $d = \Delta(C)$
- alphabet size $q = |\Sigma|$

Linear code: $\Sigma = \mathbb{F}_q$ $\quad C \subseteq \mathbb{F}_q^n$ linear subspace

Generator matrix $\quad G \in \mathbb{F}_q^{k \times n}$ $\quad$ Notation $[n,k,d]_q$

Rows of G = basis of C

$$C = \{ \alpha G \mid \alpha \in \mathbb{F}_q^k \}$$ (can think of α as the actual messages)

Parity check matrix $\quad H \in \mathbb{F}_q^{n \times (n-k)}$

$$C = \{ y \in \mathbb{F}_q^n \mid yH = 0 \}$$

Both are compact representations of code C
Computationally (essentially) equivalent

HAMMING CODE

$\forall \ell \in \mathbb{N}^+$ exists

$[2^\ell - 1, 2^\ell - \ell - 1, 3]_2$ - code

Code with parity checkmatrix H having rows

$1, 2, \ldots, 2^\ell - 2, 2^\ell - 1$

written in binary

Add $\ell = \Theta(\log k)$ extra parity bits to message to get distance 3 $\Leftrightarrow$ correct a single error.

This is necessary

$$B_q(y, r) = \{ x \in \mathbb{F}_q^n \mid \Delta(x, y) \leq r \}$$

Hamming ball. Volume of this ball

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

THEOREM 6 (HAMMING)

If an $(n, k, d)_q$ - code exists, then

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

Codes that meet this bound are called perfect

Hamming codes are perfect.

Only two other codes perfect

So Hamming bound rules out $(n, n-d+1, d)_2$ - code already for $d=3$ in binary case

But why did we ask only for $k = n-d+1$? Why not be even more ambitious (for large alphabets, say)? Because this is an upper bound regardless of alphabet

THEOREM 8 (SINGLETON BOUND)
If C is an $(n,k,d)_q$ - code, then $d \le n-k+1$.

Proof Let Σ alphabet of C. Consider projection map $\pi: \Sigma^n \to \Sigma^{k-1}$ that projects every word $\in \Sigma^n$ to its first $k-1$ coordinates.

Since $|\mathrm{Range}(\pi)| = q^{k-1}$
$|C| = q^k$

There exist $x, y \in C$, $x \neq y$, such that $\pi(x) = \pi(y)$. I.e., x and y agree in first $k-1$ coordinates.

Then $\Delta(C) \le \Delta(x,y) \le n-(k-1)$ $\boxed{\text{Ra}}$

DEF 9
An $(n,k,d)_q$ - code C is a MAXIMUM DISTANCE SEPARABLE (MDS) CODE if C meets the Singleton bound, i.e., if $k = n-d+1$
$(d = n-k+1)$

MDS codes and perfect codes are incomparable. (i.e. $\exists$ non-perfect MDS codes and $\exists$ non-MDS perfect codes)

We will next study an important family of MDS codes

REED-SOLOMON CODES (RS CODES)

Introduced in [Reed & Solomon 1959]

Constructed using univariate polynomials.

Relies heavily on following basic property

FACT 10 (DEGREE MANTRA*)

Two distinct polynomials $p_1, p_2 \in \mathbb{F}_q[x]$ of degree strictly less than k agree in strictly less than k points in $\mathbb{F}_q$. That is, there are at most $k-1$ distinct points $\alpha \in \mathbb{F}_q$ such that $p_1(\alpha) = p_2(\alpha)$.

Proof $p_1 - p_2$ is a non-zero polynomial of degree $< k$ and hence has $< k$ roots. ▨

For ease of exposition, describe RS codes by encoding function

$RS_{q,n,k}$ code

q = alphabet size = $|\mathbb{F}_q|$

$n \leq q$ block length

$k \leq n$ message length

---

*) As referred to by Ryan O'Donnell

$RS_{q,n,k}$ is constructed as follows

[(0) Generate field $\mathbb{F}_q$ explicitly (so that we can work in it — we'll ignore this issue)]

(1) Pick n distinct elements $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$ (requires $n \leq q$)

(2) Let the message $\vec{c} \in \mathbb{F}_q^k$ be $c_0, c_1, \ldots, c_{k-1} \in \mathbb{F}_q$ and identify it with polynomial
$$P(x) = \sum_{j=0}^{k-1} c_j x^j$$

(3) Encode $\vec{c} = P(x)$ by evaluation at all α_i, i.e., the codeword is
$$E(P(x)) = (P(\alpha_1), P(\alpha_2), \ldots, P(\alpha_n))$$

Clearly, $RS_{q,n,k}$ is linear code
$$\alpha \cdot E(P(x)) = E(\alpha \cdot P(x))$$
$$E(P(x)) + E(Q(x)) = E((P+Q)(x))$$

Also clear that block length $= n$, message length $= k$.
Need to argue distance

PROPOSITION 11 $RS_{q,n,k}$ is an $[n, k, n-k+1]_q$-code

Proof

Note that this meets the Singleton bound, and so is best possible. Hence $d \leq n-k+1$, and we only need to prove $d \geq n-k+1$.

Suppose we have two distinct codewords
$E(P(x))$ and $E(Q(x))$
$\deg(P(x)), \deg(Q(x)) < k$
By the Degree mantra they agree on at most $k-1$ points of $\mathbb{F}_q$ $\Rightarrow$
$P(x)$ and $Q(x)$ disagree on at least $n-k+1$ of the points $\{\alpha_1, \ldots, \alpha_n\}$ $\boxed{}$

We have the following theorem

THEOREM 12
For every prime power q and every pair $k, n \in \mathbb{N}^+$ such that $k \le n \le q$ there exists an $[n, k, n-k+1]_q$ - code.

RS codes are possibly the most commonly used codes in practice.

Used on CDs and DVDs

Can be decoded efficiently — will talk about this later.

How? This is a code over $\mathbb{F}_q$ for q large.

Embed $\log_2 q$ bits as element $\alpha \in \mathbb{F}_q$

Yields $(n\lceil \log_2 q \rceil, k \log_2 q, n-k+1)_2$ - code.

Not too bad code, but smarter constructions could do better (= get larger distance)

But... Error correction will be much better if errors occur in <u>bursts</u>. Even if $\log_2 q$ bits are corrupted, as long as they are adjacent this is 1-2 positions

So this code is good at correcting burst errors. And apparently typical errors on storage devices tend to happen in bursts

Bottle-neck of RS codes: $q \geq n$
Can get around this by using multivariate polynomials

REED-MULLER CODES (RM CODES)

Look at <u>total degree</u> of multivariate polynomials

<u>Ex</u> $x^2 y^3 z + x^4 z$ has total degree 6

$RM_{m,l,q}$ code

Alphabet $\mathbb{F}_q$ q prime power

Total degree of polynomials l
m-variate polynomials over $x_1, x_2, \ldots, x_m$

Two cases depending on relation between l and q

Case 1 $\ell < q$

Message sequence of coefficients
$$(m_{i_1,\dots,i_m})_{i_1+\dots+i_m \le \ell}$$

Represents polynomial
$$M(x_1,\dots,x_m) = \sum_{i_1+\dots+i_m \le \ell} m_{i_1,\dots,i_m} x_1^{i_1} \cdots x_m^{i_m}$$

Codeword
$$(M(\alpha))_{\alpha \in \mathbb{F}_q^m}$$

Block length $n = q^m$

Message length # m-long sequences of non-negative integers that sum to at most ℓ
$$= \binom{m+\ell}{m}$$

Example $m = 3$, $\ell = 4$

Place $m+\ell$ balls ○ ○ ○ ○ ○ ○ ○

Colour m balls ○ ○ ● ● ○ ● ○

i_j = # white balls between $(j-1)$st and jth coloured ball

$i_1 = 2$ $i_2 = 0$ $i_3 = 1$

$$\sum_j i_j = 3 \le 4$$

What about distance?

It is $\left(1 - \frac{\ell}{q}\right) n$

Follows from generalization of Degree mantra

SCHWARTZ-ZIPPEL (-DEMILLO-LIPTON) LEMMA 13

A non-zero polynomial $f \in \mathbb{F}_q[x_1, \ldots, x_m]$ of total degree d is zero on at most a $\frac{d}{q}$ fraction of the points in $\mathbb{F}_q^m$

For different messages $M \neq M'$
$M - M'$ is a non-zero polynomial of total degree $\ell \Rightarrow$ zero on at most $m \cdot \frac{d}{q}$ points.

Encodings of M and M' differ in
$\left(1 - \frac{d}{q}\right) \cancel{q^m}\, n$ points

Proof of Schwartz-Zippel

Induction over m. Prove that for random $a_1, \ldots, a_m \in \mathbb{F}_q$ $\Pr[f(a_1, \ldots, a_m) \neq 0] \geq 1 - \frac{d}{q}$.

Base case: f univariate — Degree mantra says at most d roots

$$\Pr[f(a) \neq 0] \geq \frac{q-d}{q} = 1 - \frac{d}{q}$$

Inductive step

Write $f(x_1, \dots, x_m) = \sum_{i=0}^{d} x_1^i f_i(x_2, \dots, x_m)$

f is non-zero, so some $f_i \neq 0$

Fix largest i^* s.t. $f_{i^*} \neq 0$

By IH

$$\Pr_{a_2, \dots, a_m}[f_{i^*}(a_2, \dots, a_m) \neq 0] \geq 1 - \frac{d - i^*}{q}$$

If $f_{i^*}(a_2, \dots, a_m) \neq 0$, then

$f(x_1, a_2, \dots, a_m)$ is a non-zero ~~univariate~~ poly of degree i^*, and so is $= 0$ for at most i values of x_1. Hence

$$\Pr[f(a_1, \dots, a_m) \neq 0] \geq$$

$$\Pr[f(a_1, \dots, a_m) \neq 0 \mid f_{i^*}(a_2, \dots, a_m) \neq 0] \cdot$$
$$\Pr[f_{i^*}(a_2, \dots, a_m) \neq 0]$$

$$\geq \left(1 - \frac{i^*}{q}\right)\left(1 - \frac{d - i^*}{q}\right)$$

$$\geq 1 - \frac{d}{q}.$$

Case 2 total degree $l > q$

Messages: Polynomials of total degree $\leq l$
Individual degree $\leq q-1$ in each variable

$$S(m, l, q) := \left\{ i_1, \dots, i_m \;\middle|\; \sum_j i_j \leq l \quad 0 \leq i_j < q \right\}$$

$$K(m, l, q) := |\, S(m, l, q) \,|$$

Messages

$$M(x_1, \dots, x_m) = \sum_{\vec{i} \in S(m, l, q)} m_{\vec{i}} \, x_1^{i_1} x_2^{i_2} \cdots x_m^{i_m}$$

Codeword $\quad \left(M(\alpha) \right)_{\alpha \in \mathbb{F}_q^m}$

Block length $\quad q^m$

Message length $\quad K(m, l, q)$

Distance

Write $\quad l = a(q-1) + b \qquad a \leq m, \; b < q-1$

Then distance is

$$\geq q^{m-a} \left(1 - \frac{b}{q} \right)$$

Follows from following variation of Schwartz-Zippel

LEMMA 14

If $f \in \mathbb{F}_q[x_1, \ldots, x_m]$ has individual degree at most s in each variable and has total degree $d = sk + r$, then f is non-zero on at least a

$$\left(1 - \frac{s}{q}\right)^k \left(1 - \frac{r}{q}\right) \quad \text{fraction}$$

of points in $\mathbb{F}_q^m$

Proof Excercise.

Example A multilinear polynomial (individual degrees ≤ 1) of total degree k is non-zero on at least 2^{-k} fraction of $\mathbb{F}_2^m$

Sample parameters for case 2 (original in Reed-Muller paper)

$q = 2$, $\ell < m$

Then $K(m, \ell, 2) = Vol_2(\ell, m) \geq \binom{m}{\ell}$

Distance $\geq 2^{m-\ell}$

$RM_{m,\ell,2}$ is a $\left[2^m, \binom{m}{\ell}, 2^{m-\ell}\right]_2$ - code

ALGEBRAIC CIRCUIT COMPLEXITY

Let's talk about randomness.
Already saw several times that randomness can be very helpful.

Examples:

(1) Good Ramsey graphs hard to construct
Random graphs (with right parameters) excellent properties

(2) Expander graphs complicated to construct
Random graphs are super good expanders

(3) Will talk later about what it means that an error correcting codes is (asymptotically) good. Tons of research has gone into code construction.
Cannot beat properties of random code over $\mathbb{F}_2 = \{0,1\}$

BPP = "P + possibility to flip random coins"

Is BPP strictly larger than P?

Actually, conventional wisdom is that probably $P = BPP$

Line of research "Hardness vs randomness"

Basic idea:

If $f: \{0,1\}^n \to \{0,1\}$

hard to compute in P, then for a random $x \in \{0,1\}^n$

$(x, f(x))$ will look like random

$(n+1)$-bit string (to poly-time algorithm)

Can stretch this to longer and longer pseudo-random strings

[Impagliazzo-Wigderson '97]

If there are explicit n-bit functions computable in deterministic time $2^{O(n)}$ that require <u>Boolean circuits</u> of size $2^{\Omega(n)}$, then $P = BPP$

Such functions are widely believed to exist

This discussion general, about Turing machines

Many problems are <u>algebraic</u> (eg problems involving polynomials)

Natural to study such problems in <u>algebraic</u> computational model

Canonical example Determinant

X $n \times n$ matrix with (i,j)th entry x_{ij}

Determinant is polynomial

$$\det(X) = \sum_{\pi \in S_n} \operatorname{sgn}(\pi) \prod_{i \in [n]} x_{i,\pi(i)}$$

S_n group of permutations $\pi : [n] \to [n]$

$\operatorname{sgn}(\pi) =$ sign of $\pi = (-1)^{\#\text{inversions of } \pi}$

Can be computed using multiplication and addition (algebraic operations) using <u>Gaussian elimination</u> in time poly(n)

Can also be computed efficiently <u>in parallel</u> on multiple computers (is in NC^2, although we don't need to know what that is)

<u>General fact</u> Any polynomial that can be computed efficiently using addition and multiplication can also be computed (even more) efficiently in parallel

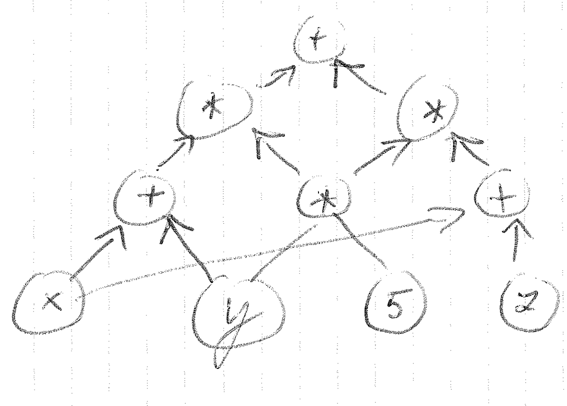
True for algebraic computation
Believed to be <u>false</u> for general Boolean computation

So extra structural restrictions can help to get stronger bounds

So algebraic computation interesting model.

In particular, we want to talk about algebraic circuits

Ex



$$(x+y)\,5y \;+\; 5y\,(x+z)$$

An algebraic circuit is a DAG
Leaves/sources labelled by
- variables x_i or
- field elements $\alpha \in \mathbb{F}$

Internal nodes/gates labelled $+$ / $*$

Size of circuit = # edges in DAG
Computes polynomials in the natural way (see example above)

Many Boolean computational tasks can also be "algebrized"
Take problem over $\{0,1\}$ and embed in $\mathbb{F}_q$

Examples
- IP = PSPACE
- Hardness of approximation of NP-complete problems (PCP theorem)

Fundamental problem in algebraic complexity theory

POLYNOMIAL IDENTITY TESTING (PIT)

Given an algebraic circuit C computing a polynomial $f(x_1, \ldots, x_n)$, is $f(\vec{x})$ the identically zero polynomial?

Randomized algorithm:
- Go to large enough field $\mathbb{F}$
- Pick random point $\vec{\alpha} \in \mathbb{F}^n$
- If $f(\vec{\alpha}) = 0$ answer yes, otherwise no.

Schwartz-Zippel lemma says this will work. But can we get a DETERMINISTIC algorithm?

Two flavours of algorithms

Black-box

Just try to find out by evaluating circuit on points in $\mathbb{F}^n$ as a black box

White-box

Also analyze structure of circuits (but might not be easy to use this info)

Black-box algorithms equivalent to finding HITTING SETS

$\mathcal{H} \subseteq \mathbb{F}^m$ is hitting set for class of circuits $\mathcal{C}$ of certain structure if $\forall$ non-zero $C \in \mathcal{C}$ $\exists\, \vec{\alpha} \in \mathcal{H}$ s.t. $C(\vec{\alpha}) \neq 0$

It can be shown that small hitting sets exist. The problem is to find small explicit hitting sets for interesting classes of circuits $\mathcal{C}$

(General case of unrestricted circuits beyond reach).

PIT is a fundamental problem that turns up in different contexts

Example Bipartite matching can be reduced to PIT

Take bipartite graph $G = (U \dot\cup V, E)$ $|U| = |V| = n$

Define matrix $\mathcal{X}$ such that

$$(\mathcal{X})_{ij} = \begin{cases} x_{ij} & \text{if edge from } i \in U \text{ to } j \in V \\ 0 & \text{otherwise} \end{cases}$$

$\det(X)$ non-zero $\Longleftrightarrow$ exist ~~complete~~ perfect matching

This shows that determining whether a bipartite graph has a perfect matching can be determined fast in parallel with randomness (is in randomized NC)

This is the only way known to prove this.

<u>Open problem</u> Construct any nontrivial hitting set $\mathcal{H}$ for this class of determinants.