Error-correcting codes

Use $\mathbb{F}_2$ as alphabet (don't need field structure except for linear codes)

code $C$ of block length $n = \text{subset of } \mathbb{F}_2^n$

rate $= \frac{\log_2 |C|}{n}$

d = min. dist. = min. Hamming dist. between dist. pts in $C$

$$d_H(x,y) = \# \{ i : x_i \neq y_i \}$$

can correct $< \frac{d}{2}$ errors

(7,4) Hamming code

\[
\begin{bmatrix}
1000000 \\
1100000 \\
1110000 \\
1110100 \\
1110110 \\
1110111 \\
1110110
\end{bmatrix}
\]

7 bits

\[
\begin{bmatrix}
1 \to \text{parity bits} \\
10 \to \text{data bits} \\
11 \to \text{parity checks (set parity bits to ensure)}
\end{bmatrix}
\]

3 parity checks

Single error-correcting (parity checks tell which bit is wrong)

algebra vs. graph theory

algorithms crucial

constructive vs. non-constructive codes

Reed-Solomon codes

block length $n$, $q > n$

$X_1, \ldots, X_n \in \mathbb{F}_q$ distinct

impractical, not common

impractical, very uncommon

written Lee dist., etc.

time vs. efficiency

deep space comm

118
codewords:
\[(f(x_1), \ldots, f(x_n))\]
f poly, \(\deg(f) \leq k\)
(uniquely defined by \(k\) pts.)
dimension \(k\) linear code, rate \(\frac{k}{n}\)
min. dist. \(n-k+1\)
(dist. polys agree at \(\leq k\) pts. \(\implies\) disagree at \(\geq n-(k-1)\) pts.)

"MDS code"
great for erasure channel
(no silent corruption; know which data missing TCP/IP)
except not efficient enough if large
(linear algebra: asymp. fast but not practical)

**Secret sharing** (Shamir)
Share a secret among \(n\) people so only \(k\) together can reconstruct
(easy: \(n=k=2\))

A random poly of \(\deg \leq k\) w/ secret = \(f(0)\)

shares = \(f(x_i)\) (\(x_i\) dist., \(\neq 0\))

< \(k\) shares \(\Rightarrow\) **nothing** about \(f(0)\)

\(k\) shares \(\Rightarrow\) determine \(f(0)\).
What about general errors (not just erasure)?
  
  Encoding easy, but decoding?

Various alg.

Berlekamp-Welch (can be made more efficient than described here)

Want to correct up to \( e < \frac{n-k+1}{2} \) errors receive \((y_1, \ldots, y_n)\)

Will find polys \( g, h \)

\[ \deg(g) = e \]
\[ \deg(h) < e+k \]

\[ \text{ s.t. } y_i \cdot g(x_i) = h(x_i) \]

Idea: \( g = \) error locator, roots at errors

\( h = g f \)

\underline{Lemma} Given \( g_1, h_1 \), \( g_2, h_2 \) as above

\[ \frac{h_1}{g_1} = \frac{h_2}{g_2} \]

\begin{proof}
\[ \deg(h_1 g_2 - g_1 h_2) \leq 2e + k - 1 < n \]
but vanishes at \( x_1, \ldots, x_n \). Q.E.D.
\end{proof}

Find \( g, h \) by solving linear eqns. Set leading coeff of \( g = 1 \).

\[ f = \frac{h}{g} \] (check if poly.)

See if this \( f \) works.

If anything fails (no \( g, h \), or \( f \) not poly., off fails)

Then \( > e \) errors.
Disadvantage of R-S. codes: 
Q must be large

but works beautifully for bursty channels
(if have one error in a field elt., might
as well have many)

⇒ widely used
(Importance of channel modeling)

Algebraic-geometric codes

X smooth proj. curve / \mathbb{F}_q
absolutely irreducible

function field K over \mathbb{F}_q

each pt. p in \overline{X}(\mathbb{F}_q) gives valuation v_p on K

|f|_p = q^{−v_p(f)}

(also valuations from Galois orbits of pts
in \overline{X}(\mathbb{F}_q^s))

S nonempty subset of \overline{X}(\mathbb{F}_q)

\Omega_S = \{f \in K : \text{all poles are in } S \}^s
Divisor D supported in \( \mathbb{X}(\mathbb{F}_q) \)
(formal \( \mathbb{Z} \)-linear comb of finitely many \( \mathbb{F}_q \) pts.)
effective \( \deg D \geq 0 \): coeff. \( \geq 0 \)
\( f \in \mathbb{K}^* : \ (f) = \sum_p \nu_p(f) \rho_p \)
principal divisor

Riemann-Roch space
\( \mathcal{L}(D) = \{ 0 \} \cup \{ f \in \mathbb{K}^* : (f) + D \geq 0 \} \)

D specifies max. order of pole at each pt.

Riemann-Roch:
\[
\dim_{\mathbb{F}_q} \mathcal{L}(D) = \deg(D) - (g-1) \\
+ \dim_{\mathbb{F}_q} \mathcal{L}(\omega - D)
\]

\[
\Rightarrow \dim_{\mathbb{F}_q} \mathcal{L}(D) \geq \deg(D) - (g-1)
\]

if \( \deg(D) > 2g-2 \)

AG code: \( \text{supp}(D) \subseteq S \)
specifies order of poles
distinct \( p_1, \ldots, p_n \) & \( S \)

codeword: \( (f(p_1), \ldots, f(p_n)) \)
\( f \in \mathcal{L}(D) \)
Example. proj. line $\mathbb{P}^1(F_8)$
\[ S = \{ \infty \} \]
\[ \mathcal{O}_S = \mathbb{F}_8[2] \]
\[ D = \text{div}(u - 1) \Rightarrow \]
\[ \mathcal{L}(D) = \text{poly} \text{ of deg} < k \] so get R.-S. codes

Min. dist. ?

Product formula
\[ f \neq 0 \implies \sum_{p \in \mathcal{D}} n_p(f) = 0 \]
(principal divisors have degree zero)

\[ \implies \text{Two elements of } \mathcal{L}(D) \text{ can agree at at most } \text{deg}(D) \text{ pts.} \]

\[ \implies \text{min. dist. } \geq n - \text{deg}(D) \]

Why AG codes? Generally not practical currently.

Beat Gilbert-Vashamov (random const.)!
One of rare cases where algebra beats prob. comb. asymptotically.
Erasure channel
TCP/IP, multicast, etc.
“forward error correction”
Luby-Mitzenmacher-Shokrollahi-Spielman-Stemann

“tornado codes”

data

parity check

6-particle graph

Simple message passing algorithm
Find degree 1 constraints + remove.
Complexity \leq \# edges in graph

Design graphs (degree distribution) to optimize performance.

Layer

Get linear-time enc., dec.
Within \epsilon of channel capacity (define!)

Super efficient in both theory + practice!
problem: need to know erasure rate
     not good for broadcast, etc.

Digital Fountain (concept + company)
rateless codes
just collect chunks until have enough to decode
applies to broadcast, backups
Suppose have $k$ packets total. Assume synchronized PRGs

1st attempt
Send one randomly selected packet each time.
coupon collector problem
$\mathbb{E} T_k = \text{exp. time to get } k$ of them all
$\text{Prob} \left( \text{don't get } i \text{ in } n \text{ trials} \right) \leq \left( 1 - \frac{1}{k} \right)^n \leq e^{-\frac{n}{k}}$

$n = k \log(k \epsilon) \Rightarrow \frac{3}{k}$

$\text{Prob} \leq \epsilon$ of missing info. in $k \log(k \epsilon)$ trails

TODO SLOW (log(k\epsilon) factor)
Luby transform

At each time step:
   choose d from degree distribution.
   Send XOR of d random data packets
   decoding: look for degree 1 Then remove
   Can achieve recovery from
   \[ k + C \sqrt{\log(M/k)} \]
   packets w/ prob 1-\epsilon and
   \[ O(k \log k) \] computation time.

Note: log k factor is necessary for LT setting.
Raptor codes (Shokrollahi)
   layer w/ fixed-rate code to remove this.

Best fountain codes known (theory + practice)

Must choose degree distribution carefully
1st attempt was all 1.
2nd attempt

Take XOR of random subset of data packets
degree dist. \( p_d = \binom{k}{d}/2^n \).

Given \( k \) received packets,

\[
\text{prob can recover} = (1-\frac{1}{2^k})(1-\frac{1}{2^{k-1}})\ldots(1-\frac{1}{2})
\]

converges to 0.288... as \( k \to \infty \)

(very good)

If receive \( n \geq k \),
look at the \( n \) coeff vectors in \( \mathbb{F}_2^k \)

\[
\text{Prob (all in a given hyperplane)} = 2^{-n}
\]

only \( 2^{k-n} \) hyperplanes \( \Rightarrow \)

\[
\text{Prob (full rank)} \geq 1 - \frac{1}{2^{n-k}}
\]

If receive \( k + \log_2 k \),

can recover w/ prob. \( \geq 1 - \varepsilon \)

and time \( O(k^3) \)

great rate, lousy complexity
ideal: want always one degree 1 vertex at each time step

Can achieve in expectation, but no good since variance ruins algorithm.

Let \( n_d(t) = \exp \# \text{deg. } d \text{ vertices at time } t \)

Time \( t \): \( k-t \) data packets remaining

Prob. a degree \( d \) will contain any given one is \( \frac{d}{k-t} \).

\[
\begin{align*}
    n_d(t+1) &= \begin{cases} 
        n_d(t)(1 - \frac{d}{k-t}) + n_{d+1}(t) \cdot \frac{d+1}{k-t} & \text{if } d > 1 \\
        n_d(t) - 1 + n_{d+1}(t) \cdot \frac{d+1}{k-t} & \text{if } d = 1
    \end{cases}
\end{align*}
\]

Solve for \( n_1(t) = 1 \) for all \( t \).

\[ \Rightarrow n_d(t) = \frac{k-t}{d(d-1)} \text{ for } d > 1. \]

So \( p_1 = \frac{1}{k} \)

\[ p_d = \frac{1}{d(d-1)} \text{ for } 2 \leq d \leq k \]

(telescoping series)

"ideal soliton distribution"

No good! Variance ruins this.
What if we replace 1 w/ 1+m?

\[ m = c \log(\frac{\nu(e)}{\sqrt{k}}) \]

to avoid fluctuations due to variance.

\[ v_d(t) = \begin{cases} 
1+m & d = 1 \\
\frac{k-t}{d(d-1)} + m & d > 1 
\end{cases} \]

exp. # packets = \( k + mH_k \)

very good

exp. # edges = \( 1 + \sum_{d=2}^{k} \frac{k}{d-1} + mk \)

\[ = 1 + k + H_{k-1} + mk \]
\[ \approx k^{3/2} \]

TOO BIG

\( \mathcal{O}(k^{3/2}) \) time alg.

final: "robust solution dist."

\[ t_d = \begin{cases} 
m \frac{m}{\nu d} & 1 \leq d < \frac{k}{\nu m} \\
m \log(m(e)) \nu & d = \frac{k}{\nu m} \\
0 & d > \frac{k}{\nu m} \end{cases} \]
Take ideal soliton \( p_d^{1/5} \).

Degree dist: \( \frac{p_d + T_d}{2} \) normalizing factor

Can show that w/ prob. \( 1-\varepsilon \),
given \( k + c \log(k)^2 \sqrt{k} \) received packets,
can decode in \( O(k \log k) \) ops.