Today: Algorithms in Coding Theory

- Algorithmic Problems
- Erasure Decoding
- Decoding Reed Solomon Codes
- Abstracting for Other Algebraic Settings

Algorithmic Problems

1. Encoding:

2. Detect Errors:

3. Correct Erasures

Complexity of 1, 2, 3.

- "Easy" for linear codes, given \( \mathbf{G} \)

**Encode:**
\[
\begin{align*}
\mathbf{m} \in \{0,1\}^k \\
\downarrow \\
\mathbf{mG} \in \{0,1\}^n
\end{align*}
\]

**Detect:** Compute \( \mathbf{H} \) s.t. \( \mathbf{G} \cdot \mathbf{H} = 0 \)

Given \( \mathbf{y} \) output \( \text{OK} \) if \( \mathbf{y} \cdot \mathbf{H} = 0 \)

**Erasure Correct:** Given \( \mathbf{r} \in \{0,1,?\}^n \), let \( \mathbf{G}' = \mathbf{G} \) with \( ? \)ed columns deleted.

\[
\mathbf{y}' = \mathbf{y} \quad \text{(?)}
\]

Then \( \mathbf{m} : \mathbf{mG}' = \mathbf{r}' \)

**Claim:** if \( \#\{?\} < \Delta(C) \) then \( \mathbf{m} \) is uniquely determined.
Subtleties?

1. Works only for linear codes...
2. Assumes \( G \) known.
3. Assumption 2 not always valid. Eg. GI bound (valid when you prove it! Construct \( b \) uniformly, efficiently)
4. Is problem well defined for non-linear codes? My “defn.”: Code is “constructive” if can construct encoding circuit \( C: \{0,1\}^k \rightarrow \{0,1\}^n \) in poly time.
Decoding?
- Not so simple -- can't seem to handle generic $g$
- Even for special $g$'s -- can only arrest limited # errors ....
- But these may be sufficient for
  \[
  \text{Hamming} / \text{Shannon} / (\text{Elia})
  \]

Shannon Problem: Given $y$ find $m$ that
maximizes $\Pr[y|m]$.
O.K. to be wrong on some $y$ ....
Provided $\Pr[y]$ (exponentially) small,

[[Average case complexity / Worst-case]]
**Hamming Problem**: Given $y$ find $m$ that

minimizes $\Delta(y, m6)$.

O.K. to be "wrong" if

$$\min_m \left\{ \Delta(y, m_6) \right\} > \frac{\Delta(c)}{2}$$

or some $\ell$

But if $\# \text{ errors} \leq \ell$, must get it right!

[[Worst-Case Complexity]]

[[if $\ell > \frac{\Delta(c)}{2}$ .... can produce small list

including every $m$ s.t. $\Delta(y, m_6) \leq \ell$]]

Seems hard, but can be done .... e.g., for RS codes

[[Peterson, Berlekamp, Massey, S., Guusuami-S.]]
Reed-Solomon Decoding

Problem

Given: RS code = (F, α, ..., α_n, k)

\[ r = (r_1, ..., r_n) \in F^n \] (List all)

Output: Deg. k-1 poly \( p(x) = \sum_{i=0}^{k-1} x^i \)

\[ \# \text{errors} = \sum | p(a_i) \neq r_i | \leq t \]

[Today: \( t \leq \frac{n-k}{2} \); so \( 0 \leq \text{list size} \leq 1 \).]

Algorithm: Peterson 1960: Defined “P”

Welch-Berlekamp 1986

Gemmell-Sudan 1992

[Kinden, gentler...]

**Key Idea:** "Error Locator Polynomial"

Define: $\text{Err} \triangleq \Sigma \{ i | p(\alpha^i) + r_i \}$

(Warning: Don't know $p(\cdot)$ & so don't know $\text{Err}$! But still ...)

$\Xi(x) \triangleq \prod_{i \in \text{Err}} (x - \alpha^i)$

(Extended Warning: Don't know $\Xi(x)$ either ...)

$\Xi(x)$ has nice properties
Properties of $E(x)$

1. $\forall i, \quad p(\alpha_i) \cdot E(\alpha_i) = r_i \cdot E(\alpha_i)$

2. $\deg E \leq t$ ; $E \neq 0$

3. $N(x) \leq p(x).E(x)$ is a poly of
   $\deg N \leq t+k-1$

1'. $\forall i, \quad N(\alpha_i) = p(\alpha_i) \cdot E(\alpha_i) = r_i \cdot E(\alpha_i)$

Algorithm: Ignore all references to "$p" above

1. find $(N, E)$ !

\underline{Step 1}: find $N, E$ s.t.

(i) $\forall i, \quad N(\alpha_i) = r_i \cdot E(\alpha_i)$

(ii) $\deg N \leq k+t-1$ ; $\deg E \leq t$ ; $E \neq 0$

\underline{Step 2}: Output $p(x) = \frac{N(x)}{E(x)}$ (if $\deg \leq k-1$ poly)
Analysis: Correctness? Efficiency?

Efficiency: Step 1: Just a big linear system

Step 3: Ratios ... Long Division.

Correctness:

Lemma 1: \( \exists (N, E) \) satisfying (i), (ii) provided

# errors in \( i \leq t \); with \( N/E = p \).

Proof: Take \( E \) to be error locator; \( N = E \cdot p \).

Lemma 2: if \( \exists \) two pairs \( (N_1, E_1) \), \( (N_2, E_2) \)

satisfying (i), (ii); then \( N_1/E_1 = N_2/E_2 \)

(provided \( N > k + 2t \)) \( \iff \) \( N_1 \cdot E_2 = N_2 \cdot E_1 \).

Proof: \( \forall i \)

\[ N_i(\alpha_i)E_2(\alpha_i) = \gamma_i E_1(\alpha_i).E_2(\alpha_i) \]

But both are deg.

\[ = E_1(\alpha_i).N_2(\alpha_i) \]

Identical \( \iff \) agree at \( n > k + 2t \) points \( \iff \) polys.
Abstraction:

Key property of polynomials:

- Product of deg $d_1, d_2$ deg poly is $d_1 + d_2$ poly.

For $U, V \in \mathbb{F}^n$:

- Let $U \ast V \triangleq (U_1 V_1, U_2 V_2, \ldots, U_n V_n)$
  = coordinate-wise product.

For sets $S, T \subseteq \mathbb{F}^n$:

- $S \ast T \triangleq \{ U \ast V \mid U \in S, V \in T \} \subseteq \mathbb{F}^n$

Key Property:

- $S = \text{RS code, dim } k$ \iff $S \ast C \leq \text{dim } k + t$
- $T = \text{RS code, dim } t + 1$ \iff $k + t$

(Generically \ldots expect \text{dim } k \cdot t)
Abstract Decoding

Given linear code \( C = [n, k, d] \), for a \( \epsilon \)-error decoding pair \((\xi, N)\) if (i) \( \dim(\xi) > \text{large}_{1} \),

(ii) distance \((N) > \text{large}_{2} \),

(iii) \( E \ast C = N \),

(iv) \( \text{dist}(\xi) > \text{large}_{3} \),

Exercise: Fill these values in

Algorithm: Given: \( r = (r_{1}, \ldots, r_{n}) \);

(i) Find \( E \in \xi; \ N \in N \); \( E \neq 0 \) s.t. \( E \ast r = N \).

(ii) Let \( y_{i} = r_{i} \) if \( E_{i} \neq 0 \); \( y_{i} = ? \) o.w. \( \text{ERASURE-DECODER}(y) \).
Claim: Yields decoding algorithms for all algebraic codes; Corrects roughly \( \frac{d}{2} \) errors.
E.g. even for BCH codes!! (Exercise)

Notes:
1) Good News: Can correct \( \frac{d}{2} \) errors in RS codes.
   (Algorithm not totally intuitive .... will see more later.)
2) Actually correct \( s \) erasures & \( t \) errors, simultaneously provided \( s + 2t < d \)
3) Can now inn. correct \( \frac{d_1}{2} \cdot \frac{d_2}{2} \) errors in concatenated codes if outer = RS of dist \( d_1 \)
   & inner = dist. \( d_2 \) code
   Will show this & do better next time.