A Crash Course on Coding Theory

Madhu Sudan
MIT

Topic: Decoding Algorithms

This lecture will focus on algorithms for decoding of *algebraic* codes.

Erasure correction problem

(Gentle introduction to errors).

Defn: Erasure channel either transmits symbol faithfully, or outputs $?$.

Erasure decoding problem:

Given: G generator for code $\mathcal{C}$.

$r_1, \ldots, r_n \in \mathbb{F}_q \cup \{?\}$.

Task: Find $c \in \mathcal{C}$ s.t.

$r_i \neq ? \quad \Rightarrow \quad r_i = c_i$

Prop: c_i unique if # $?$'s is less than d.

Erasure correction (contd).

Alg:

- Delete rows of G corresponding to $?$s. Call resulting matrix G'.
- Let r with $?$s deleted be r'.
- Find x s.t. $xG' = r'$ by solving linear system.
- Output c if unique
  Else, output A, c s.t. $c + yA$ are *all* the solutions.

Conclusion:

- Erasure decoding easy for linear codes.
  - Can find soln. if unique.
  - Can enumerate all if not!

The Error Correction Problem

(Welcome to the real world.)

Task:
(Implicitly given) Code $\mathcal{C}$.
Explicit Input: $r = \langle r_1, \ldots, r_n \rangle \in \mathbb{F}_q^n$.
Parameter: Integer e.
Goal: Compute $c \in \mathcal{C}$ s.t. $\Delta(r, c) \le e$.

Error correction radius

Combinatorial question:

When is c uniquely specified (by r, e and $\mathcal{C}$)?

Prop: If $e < d(\mathcal{C})/2$ then at most one c.

(Maybe none!)

Food for thought: Which comes first? Error-correction radius? or distance? (I.e., which one to optimize, given rate?)

Answer: Doesn't matter - they are essentially optimized simultaneously!

Decoding Reed Solomon Codes

Problem Statement

Given:

- $x_1, \ldots, x_n \in F$ distinct.
- $r_1, \ldots, r_n \in F$.
- Integers k, e

Task: Find a poly p of deg. $k-1$ s.t.
$p(x_i) \neq r_i$
for at most e values of $i \in \{1, \ldots, n\}$.

Decoding Reed Solomon Codes

[Peterson60, Berlekamp66, Massey66]
[Welch-Bkmp86, Gemmell-S.92]

Key concept: Error locator polynomial

$$Y(x) \text{ s.t. } Y(x_i) = 0 \text{ if } p(x_i) \neq r_i$$

1. Y has low-degree ($\leq e$)

2. $Z = Y.p$ has low-degree ($\leq e + k - 1$)

3. $\forall i, \quad Z(x_i) = Y(x_i).p(x_i) = Y(x_i).r_i$

Main Idea: Ignore all references to p above and look for Y, Z.

Decoding RS Codes (contd.)

I. Find (Y, Z) s.t.

- $Y \not\equiv 0$
- $\deg Y \leq e$
- $\deg Z \leq e + k - 1$
- $\forall i, \quad Z(x_i) = Y(x_i).r_i$

II. Output $\frac{Z(x)}{Y(x)}$.

Demystifying Step I: Just linear algebra!

Why does it work?

Claim 1: Pair of polynomials Y, Z satisfying the requirements of Step I do exist!

(In fact we just proved the existence.)

Claim 2: Linear Algebra can find one such pair.

(But pair may not be unique. How do we guarantee Y is the error-locator?)

Claim 3: If Y, Z and Y', Z' both satisfy conditions of Step I, then $Z/Y \equiv Z'/Y'$.

Proof of Claim 3

Consider the polynomials $Y' \cdot Z$ and $Y \cdot Z'$.

- Both have deg. $\leq 2e + k - 1$.
- For every $i \in \{1, \ldots, n\}$,
  $Z(x_i) = Y(x_i) \cdot r_i$ and $Y'(x_i) r_i = Z'(x_i)$.
- Multiplying and cancelling r_i's:
  $(Y' \cdot Z)(x_i) = (Y \cdot Z')(x_i)$.
- But above happens for n points, while degrees are smaller than n!
- So $Y' \cdot Z \equiv Y \cdot Z'$

Thm: Alg. works if $e \leq \frac{n-k}{2}$.

(As given, runs in time $O(n^3)$ time. Best implementations take $O(n \operatorname{poly} \log n)$.)

Musings

- Algorithm essentially in [Peterson'60]. Before "polytime" was formalized.
- Magic of algebra! Also a warning shot! Beware if you intend to base cryptography on algebra ...
- Roots of the specific algorithm. CS literature: [Berlekamp-Welch'86]. All ideas are there, but not the exposition. Exposition is from [Gemmell-S.'92].
- But equally simple exposition well-known in coding theory (from around 1988). [Pellikaan,Kotter,Duursma].
- We'll describe their knowledge next.

Abstract decoding algorithm

- How much of the prev. algorithm is linear algebra? And how much polynomial arithmetic?
- Investigated by [Pellikaan,Kotter,Duursma 88].
- Surprisingly little polynomial arithmetic.

Abstract decoding (contd.)

Fix a code $\mathcal{C} = [n, k, d]$.

Defn: $(\mathcal{Y}, \mathcal{Z})$ are e-error-correcting pair for $\mathcal{C}$ if the following hold:

- $\mathcal{Y}$ are linear codes.
- $\mathcal{Y} = [n, e+1, n-d+1]$ code.
- $\mathcal{Z} = [n, ?, e+1]$ code.
- $\mathcal{Y} * \mathcal{C} \subset \mathcal{Z}$, where

$A * B = \{a * b | a \in A, b \in B\}$ and $a * b$ denotes coordinatewise product.

Thm: If $\mathcal{C}$ has a e-error-correcting pair then it has an e-error-correcting algorithm.

Algorithm

Given: $r = \langle r_1, \ldots, r_n \rangle \in \mathbb{F}_q^n$.

- Find $(y \in \mathcal{Y}, z \in \mathcal{Z})$ s.t.
  - $y \neq 0$.
  - $y * r = z$.
- Set $c_i = r_i$ if $y_i \neq 0$ and erasure otherwise.
- Erasure decode for c.

Proof steps

1. Such a pair (y, z) exists:
   - Set y_i to zero whenever $c_i \neq r_i$.
   - Find non-zero $y \in \mathcal{Y}$ subject to above. (Exists by dim. of $\mathcal{Y}$.)
   - Set $z = c * y$.

2. Pair can be found (linear system).

3. For any (y, z) found by alg. and any c s.t. $\Delta(c, r) \leq e$, we have $y * c = z$. (Follows from distance of $\mathcal{Z}$.)

4. Any pair y, z has at most one c s.t. $y * c = z$. (Follows from distance of $\mathcal{Y}$.)

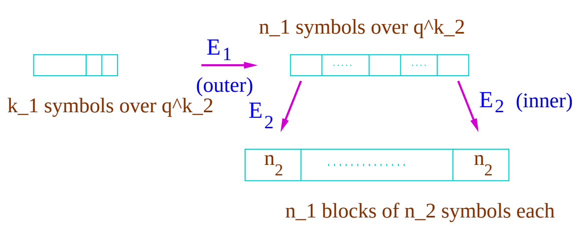
Application: AG codes

- Recall order axioms for algebraic-geometry codes. (Product rule, and # zeroes.)
- $\mathcal{C} =$ functions of order $< k$..
- $\mathcal{Y} =$ functions of order $< (n - k + g)/2$.
- $\mathcal{Z} =$ functions of order $< (n + k + g)/2$.
- Gives $(n - k - g)/2$-error-correcting pair.
- Thus every AG code $\mathcal{C}$ has a decoding alg. going up to $(d(\mathcal{C}) - g)/2$ errors.

Decoding Concatenated Codes

Recall concatenation:

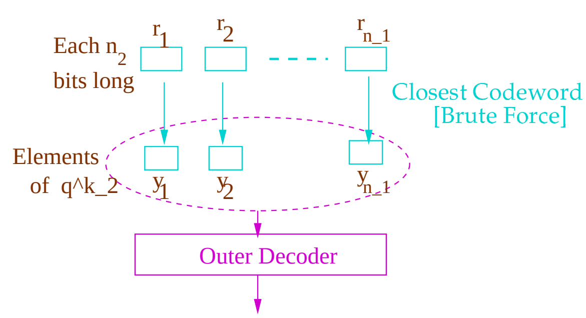
$$[n_1, k_1, d_1]_{q^{k_2}} \circ [n_2, k_2, d_2]_q$$



[Forney'66]: Also gave decoding algorithms.

Simple decoding

Prop: If outer code decodable up to e_1 errors (in poly time), then concatenated code is decodable up to $e_1 \cdot \frac{d_2}{2}$ errors in poly $+O(n_1 q^{k_2})$ time.



Alg: Decode each symbol of inner code by Brute force. Then decode the "received word" corr. to outer code.

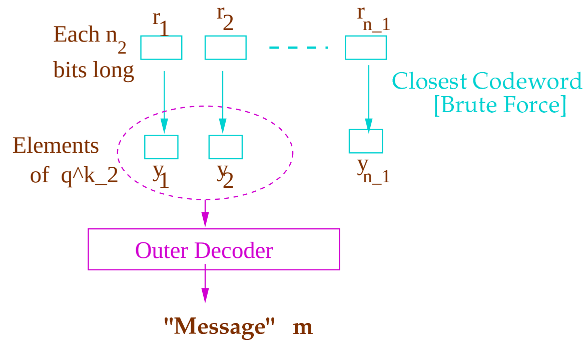
Generalized Min. Dist. Decoding

More sophisticated decoding. Stronger assumptions. Stronger result. $[Forney]$.

Assumption: Outer code has error and erasure decoder. Decodes if $2e + s < d_1$, where $e = \#$ errors, $s = \#$ erasures.

Consequence: Concat. code can be decoded for up to $d_1 d_2/2$ errors ($=$ half the minimum distance).

GMD Algorithm



Alg:

- Let $w_i = \min\{\Delta(r_i, y_i), d_1/2\}$.
- W.l.o.g. $w_1 \le w_2 \le \cdots \le w_{n_1}$.
- For $i = 1$ to n_1 do
  - Declare $\{i, \ldots, n_1\}$ to be erasures.
  - Decode prefix.

GMD Analysis

- Let m be s.t. $\Delta(E_2(E_1(m)), r) < d_1 d_2/2$.
  Let $\langle z_1, \ldots, z_{n_1}\rangle = E_1(m)$.
  Let $l_i = \Delta(z_i, r_i)$.
  Let $b_i = 1$ if $z_i \ne y_i$.
- Assume decoding unsuccessful. Then following inequalities hold:
  (1) $\forall j, \quad (n_1 - j) + 2 \cdot \sum_{i=1}^{j} b_i \ge d_1$
  (2) $\forall i, \quad l_i \ge \max\{w_i, b_i(d_2 - w_i)\}$
  (3) $\forall i, \quad w_i \le w_{i+1} \le d_2/2$
- Above imply:
  $\sum_{i=1}^{n_1} l_i \ge \frac{d_1 d_2}{2}$

Analysis (details)

(2) $\quad \Rightarrow \quad l_i \ge w_i + b_i(d_2 - 2w_i)$

So suffices to show:
$\sum_i w_i/d_2 + \sum_i b_i(1 - 2(w_i/d_2)) \ge d_1/2$.

- Let $x_i = 1 - 2w_i/d_2$.
- Then x_i's are non-increasing, with $0 \le x_i \le 1$.
- Suffices to show:
  $\sum_i (1 - x_i/2) + \sum_i b_i x_i \ge d_1/2$,
  given $(n_1 - j)/2 + \cdot \sum_{i=1}^{j} b_i \ge d_1/2$
- Above follows if the vector
  $\langle x_1, \ldots, x_{n_1}, -\sum_i x_i\rangle$
  is in the convex hull of the vectors
  $v_1, \ldots, v_{n_1}$, where $v_j = \langle 1^j 0^{n_1 - j}, (-j)\rangle$.
- Last is easily verified.

Summarizing

- Can decode Reed-Solomon codes efficiently, up to half the minimum distance.
- Can decode algebraic codes efficiently, up to some close approximation to half the distance.
- Can decode concatenated codes also up to half the distance, provided outer code is nicely decodable.
- Why half the distance?
  - Algorithmic limitation? (Can't handle more errors?)
  - Combinatorial limitation? (Answer is not unique!)