# A Crash Course on Coding Theory

Madhu Sudan
MIT



## Topic: Decoding more errors

In this lecture we will see how we can "correct" more errors.

More correctly, we will redefine the "correction" problem, and then how to correct more errors in this definition.



## List Decoding

When # errors more than the error-correction radius, the transmitted word is not specified uniquely by the received word! What should an error-correction algorithm do?

List Decoding Problem:
Given: $r \in \mathbb{F}_q^n$, error bound $e$.
Task: Output list of codewords $c$ s.t.
$\Delta(r, c) \le e$.

- Set more appropriate than list?
- Problem dates back to [Elias '57].
  Also [Wozencraft '58].
- Reasonable notion of recovery.
- Error probabilistic $\Rightarrow$ List size $\le 1$ w.h.p.
- Clean separation of algorithmic problem from probability.



## List decoding radius

Prereq. for polytime soln. (in input):
Output size must be polynomial.

Informal Defn: The list-decoding radius (LDR) of a code $\{\mathcal{C}\}$ is the largest radius for which the output size of the list decoding problem is bounded by a polynomial in the block length.

Notes:

- Sound familiar? Recall Johnson bound!
- To formalize defn.,
  Either fix list-size (e.g. $\text{poly}(n, q) = nq$)
  Or consider infinite families of codes.
- Better to work with ratios (LDR/$n$).
- List-size $1$ gives error correction radius!

## List decoding radius

- Well understood as function of distance, for list-size $= 1$.
- Not well understood for list-size $\gg 1$.
- Fix $q$-ary $\mathcal{C}$ with dist. $\delta n$, and LDR (for poly-sized lists) is $\epsilon n$.
  - For "natural" codes $\epsilon \leq \delta$.
  - For all codes $\delta/2 \leq \epsilon$.
  - $1 - \sqrt{1-\delta} \leq \epsilon$. (Johnson bound.)
  - $\frac{q-1}{q}(1 - \sqrt{1 - \frac{q}{q-1}\delta}) \leq \epsilon$. (Johnson.)

    (Lower bounds on $\epsilon$ increasing.)



## List decoding radius

- Relation to rate of the code?
- Intuitively, LDR drops as rate increases. What is largest LDR for code of rate $R$?
- Not known if list-size $= 1$. (Why?)
- Very well understood for list-size $\gg 1$.
  - $\epsilon \leq 1 - H_q(1-R)$
  - There exist codes s.t. $\epsilon = 1 - H_q(1-R)$.
- Musing: Which is more important? LDR or Distance? If former, then why pursue all the bounds?

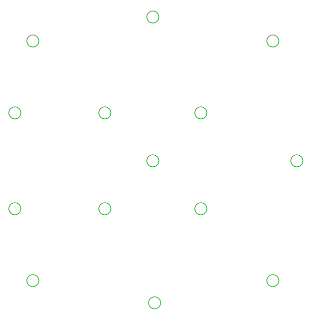


# Algorithms for List Decoding

## List decoding of Reed Solomon Codes

Given: $t, d$; $\{(x_1, y_1), \ldots, (x_n, y_n)\}$.
Task: Find all polynomials $p$ of degree $\leq d$ that go through at least $t$ points.

## Example Instance



$$n = 14 \ ; \qquad d = 1 \ ; \qquad t = 5$$



## Solution Idea

- Somehow need to explain both the curve $y = x$ and $y = -x$. Further there is the mess of points (which fortunately lie on a circle $y^2 + x^2 = 1$. Need an error-locator for this!
- How to explain all these points together? Algebraically?
- Previously: Gave "functional" explanation. $y_i = R(x_i)$ for some rational function $R$.

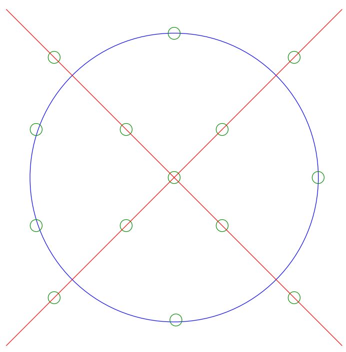


## Solution Idea (contd.)

- New idea: Try relational explanation.
- I.e. Find non-zero $Q(x, y)$ such that $Q(x_i, y_i) = 0$ for every $i$. (Ideally degree of $Q$ small in both variables.)
- Hopefully $p$ emerges in picture!
- Formally $(y - p(x)) \mid Q(x, y)$.



## A Quartic Fit: All the Lines Emerge



$$Q(x, y) = y^4 - x^4 + y^2 - x^2$$

After factoring

$$Q(x, y) = (y^2 + x^2 - 1)(y + x)(y - x)$$

## Finding $Q(x, y)$ with low degree

- Q1: How to find $Q$ of low degree?
- Q2: Why does it even exist?
- A1: Solve linear system. Let $Q(x, y) = \sum_{j,l} q_{jl} x^j y^l$.
  **Unknowns:** $\{q_{jl} : 0 \leq j, l \leq D\}$.
  **Constraints:** $\sum_{j,l} q_{jl} \; x_i^j \; y_i^l = 0$; $i \in \{1, \ldots, n\}$.
- A2: Non-zero solution exists if

$$\#\text{ unknowns} > \#\text{ constraints}$$
$$(D + 1)^2 > n$$



## Conditions for divisibility of $Q$

- $\deg_x(Q), \deg_y(Q)$ small.
- $\deg(p)$ small.
- \# pts. s.t. $Q(x_i, y_i) = y_i - p(x_i) = 0$ large.

Lemma: $\deg_x(Q), \deg_y(Q) \leq D$, $\deg(p) \leq d$, $t \geq D(d + 1)$ implies $y - p(x)$ divides $Q(x, y)$.

Why? Similar to the fact that two degree $k$ polynomials can not agree in $k + 1$ places.



## Proof

- Let $S = \{i | y_i = p(x_i)\}$.
- Let $g(x) = Q(x, p(x))$.
- $\deg(g) \leq (d + 1)D$.
- But $g(x_i) = Q(x_i, p(x_i)) = Q(x_i, y_i) = 0$ for every $i \in S$.
- $|S| > (d + 1)D$ implies $g \equiv 0$.
- Thus $p(x)$ is a root of $Q(x, y)$ (or $y - p(x)$ divides $Q(x, y)$).



## Summary

Algorithm:

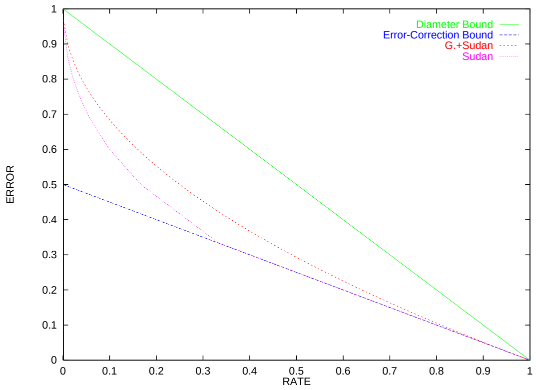
**Step 1:** Find $Q \not\equiv 0$ of degree $\sqrt{n}$ s.t. $Q(x_i, y_i) = 0$ for every $i \in \{1, \ldots, n\}$.
(Solve a linear system.)

**Step 2:** Factor $Q(x, y)$ and report all $p$ such that $y - p(x)$ divides $Q$.
(Factoring easy [Kaltofen,Grigoriev,Lenstra].

Theorem: Can find every $p$ with agreement $t$, provided $t \geq (d + 1)\sqrt{n}$.

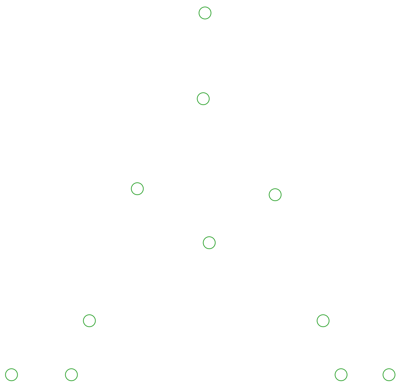
Improvements:
– Careful choice of $Q \Rightarrow t \geq \sqrt{2dn}$.
(Will do better later.)

## Algorithms & Their Performance



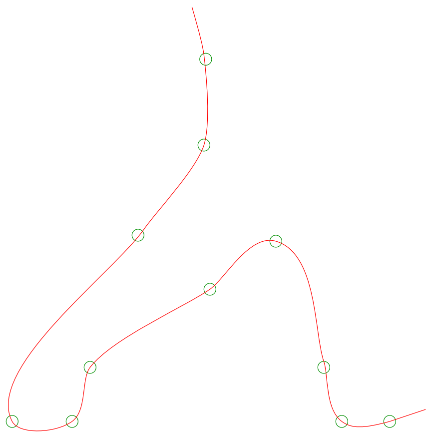



## Another Example





## A Quartic Fit: No information!!





## Why?

- Each point is good for at least two lines.
- But $Q$ has allowance to explain only one of each (i.e., is forced to go through each point only once)!
- Solution idea:
  - Similar Plan: Fit $Q$, Hope for factors of form $y - p(x)$ for all "relevant" $p$.
  - Main Difference: Expect more from $Q$.
  - Every $(x_i, y_i)$ is a "singularity" of $Q$. (Expressible as $3$ linear constraints on coefficients of $Q$.)

## Why this will help?

What we lose? – Additional constraints force us to raise the degree of $Q$. (We need more unknowns to guarantee existence of non-trivial solution.)

But....

Much more gain in the second phase when we factor $Q$ and look for factors of the form $y - p(x)$ !

- Now $p$ passes through singularities of $Q$ as opposed to just points on $Q$.
- Need only half as many singularities as regular points!!



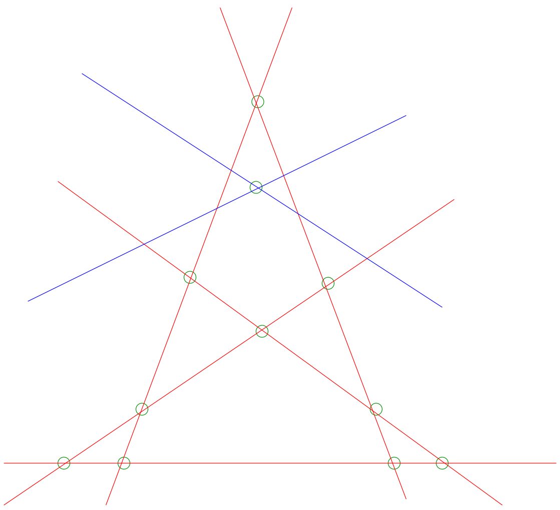
## Back to the Example

Fit $Q(x, y)$ of total degree $7$ through the points s.t $Q$ "intersects itself" at each of the $11$ points. (Can do this: solve homogeneous linear system with $\binom{7+2}{2} = 36$ unknowns and $11 \times 3 = 33$ constraints.)

Now all relevant linear polynomials "emerge" in the picture!!



## Degree 7 Super-fit: All Lines Emerge!!





## Weighted Polynomial Reconsruction Problem

Given: Integers $t, d$.
$n$ points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$
weights $w_1, \ldots, w_n$.
Task: Find all polynomials $p$ of degree $\leq d$ that go through points whose weights sum to at least $W$.

- Solution: $Q$ should have $w_i K$ singularities at $i$-th point.
- Thm: Can solve above if $W > \sqrt{d \sum_i w_i^2}$.
- $w_i = 1$ yields unweighted case.

## Application: Decoding Concatenated Codes

- Say have a concatenation of RS code with some inner code.
- New decoding approach:
  - By brute-force, list-decode inner code with distance information.
  - Convert distances into weights.
  - Apply weighted list-decoding algorithm.
- Concatenation of RS + Hadamard code can be decoded up to known bound on LDR, using above paradigm.



# Ideal Error Correcting Codes

## Some abstract algebra

Recall: Commutative rings, Integral domains.

Ideals in rings:
Defn: $I \subseteq R$ is a ideal if
$\forall \ a, b \in I, r \in R \quad a + b, ar \in I$

Examples:
- $(p) = \{pr | r \in R\}$ where $p \in R$
  Set of multiples of $p$.
- $(x, y) = (x) + (y)$ where $x, y \in R$
  Polynomials in $x, y$ with constant term $0$.

Nice Properties:
- $I, J$ ideals $\Rightarrow$ so are $I + J, I \cap J, I \cdot J$.
- $I$ factors $R$ nicely. Can define $r$ mod $I$.
  $R/I$ also a ring (and often a field).



## Abstraction: Ideals

Defn: Code specified by
- Integral domain $\mathcal{R}$,
- $n$ ideals $I_1, \ldots, I_n$, and
- Message space $\mathcal{M} \subseteq \mathcal{R}$.

$$m \mapsto \langle m(I_1), \ldots, m(I_n) \rangle$$

- Usually have norm on $R$.
- Axiom: No small non-zero element in product of too many ideals.

## Examples

Reed Solomon codes:

- $\mathcal{R} = \mathbb{F}_q[x]$
- $I_i = (x - x_i), x_i \in \mathbb{F}_q$
- $\mathcal{M} = \{p \in \mathbb{F}_q[x], \deg p < k\}$

Chinese Remainder codes:

- $\mathcal{R} = \mathbb{Z}$
- $I_i = (p_i), p_i$ prime
- $\mathcal{M} = \{0, \ldots, K-1\}$

Algebraic Geometry codes:

- Underlying field $K = \mathbb{F}_q[x_1, \ldots, x_m]/I$, $I$ some 1-dimensional ideal.
- $\mathcal{R} =$ polynomials from $K$ to $\mathbb{F}_q$.
- $I_i = P_i$ place of degree 1.
- $\mathcal{M} =$ functions of pole order $< k$.



## The decoding question

Standard definition:

Given: $r_1, \ldots, r_n \in \mathcal{R}$; $t$.
Find: $m \in \mathcal{M}$ s.t. $m \in r_i + I_i$
for at least $t$ values of $i$.

Strategy to solution:

- Redefine problem by working over $\mathcal{R}[y]$.
- Define ideals $J_i \subseteq \mathcal{R}[y]$, $J_i = I_i + (y - r_i)$.
- Note $m \in r_i + I_i \Leftrightarrow (y - m) \in J_i$.

New Problem definition:

Given: $J_1, \ldots, J_n$ ideals in $\mathcal{R}[y]$.
Find: $m \in \mathcal{M}$ s.t. $y - m \in J_i$
for at least $t$ values of $i$.



## Solution

Note: Norm on $R$ induces norm on $R[y]$

**Step 0:** Pick $z_1, \ldots, z_n$ carefully.

**Step 1:** Find $Q \in \mathcal{R}[y]$ of small norm s.t.

$$Q \in \prod_i J_i^{z_i}$$

**Step 2:** Factor $Q$ and output factors of the form $y - m$.

---

- Choice of $z_i$'s depends on $I_i$'s.
- Bound on norm of $Q$ depends on $\mathcal{M}$.
- Existence of small $Q$: counting argument.
- Finding: algorithmic question about $R$.
- $y - m | Q$ since $Q(m)$ in product of too many ideals.



## Application: Chinese remainder decoding

Given: $p_1, \ldots, p_n$ relatively prime.
$r_1, \ldots, r_n$ residues; error bound $e$.
Find: $m \in \{0, \ldots, K-1\}$ s.t.
$m \neq r_i \mod p_i$ for $\leq e$ choices of $i$.

Note: $I_i = (p_i)$, and $J_i = (p_i) + (y - r_i)$.

## CRT decoding

- Obvious norm on integers.
- Ideals $\equiv$ Lattices.
- Product $=$ intersection, if ideals coprime.
- Small polynomials found by LLL algorithm.

Algorithm:

- $L_i =$ lattice corr. to polynomials in $J_i^{z_i}$.
- $L =$ intersection of lattices.
- Find small vector $q_0, \ldots, q_d$ in $L$.
- Factor polynomial $Q(x) = \sum_i q_i y^i$ and output roots.



## References

- RS codes: [S.'97, Guruswami+S.99,]
  Based on prior work of
  [Ar+Lipton+Rubinfeld+S. '00].
- AG codes.
  [Shokrollahi+Wasserman'98, Guruswami+S.99,]
- CRT codes.
  [Goldreich+Ron+S.'98, Boneh '00]
  [Guruswami+Sahai+S. '00]
- Abstraction:
  [Guruswami+Sahai+S. '00]