Error-Correcting Codes and List Decoding

Lecture 11

Outline

Codes, distance, decoding

How good can codes be?

Reed–Solomon codes

Unique decoding: Welch–Berlekamp

List decoding

Outline

Codes, distance, decoding

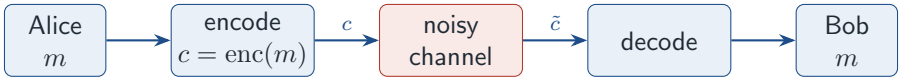
How good can codes be?

Reed–Solomon codes

Unique decoding: Welch–Berlekamp

List decoding

Communication over a noisy channel



- Alice has $m \in \{0,1\}^n$; Bob must recover it *exactly*
- the channel is analog and noisy — yet files arrive intact every day

How? Add redundancy: send a longer *codeword* $c \in \{0,1\}^N$.

The naive solution is expensive

Repetition code: repeat every bit k times (k odd), decode by majority.

m: 1 0 1 $\longrightarrow$ $\underbrace{111}\,\underbrace{000}\,\underbrace{111}$

101 000 111 received

- to get the *whole* message right with probability $\geq 2/3$ one needs $k = \Theta(\log n)$
- rate $= 1/k \to 0$: the longer the message, the more redundancy

Goal: constant rate *and* a constant fraction of errors corrected.

Worst-case errors

Model for today

an adversary, knowing everything, corrupts up to pN of the N transmitted symbols

- stronger than random noise: correcting pN worst-case errors also handles i.i.d. errors at any rate $p' < p$ w.h.p.
- with random errors alone one can sometimes tolerate a larger fraction
- both models are central; codes are also used far beyond communication — hardness amplification, pseudorandom generators, complexity

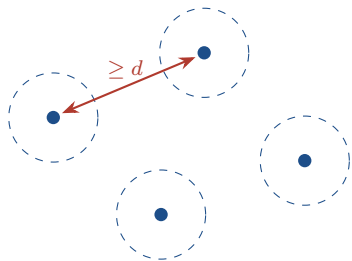
Codes, distance, rate

Definition

A code is a subset $\mathcal{C} \subseteq \Sigma^N$.

- **distance:** $d = \min_{c \neq c' \in \mathcal{C}} d(c, c')$
- **message length:** $\log_2 |\mathcal{C}|$
- **rate:** $R = \dfrac{\log_2 |\mathcal{C}|}{N \log_2 |\Sigma|}$

$d(\cdot, \cdot)$ is the Hamming distance; $\mathrm{enc} : \{0,1\}^n \to \mathcal{C}$ is any bijection onto $\mathcal{C}$.

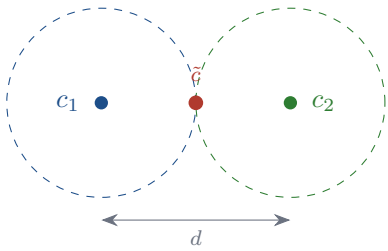


codewords, with disjoint balls of radius $\lfloor \frac{d-1}{2} \rfloor$

Distance $d \Rightarrow$ correct fewer than $d/2$ errors

- balls of radius $\lfloor \frac{d-1}{2} \rfloor$ around codewords are disjoint
- so the received word determines c uniquely: output the nearest codeword
- at $d/2$ this stops: a word halfway between two codewords has no unique answer

for now, "decode" means enumerating all of $\mathcal{C}$ — exponential time



Outline

Codes, distance, decoding

How good can codes be?

Reed–Solomon codes

Unique decoding: Welch–Berlekamp

List decoding

Random codes reach the Gilbert–Varshamov bound

Theorem (Gilbert–Varshamov)

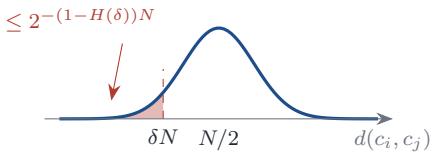
For every $\delta \in (0, \frac{1}{2})$ and $\varepsilon > 0$ and all large N, there is a binary code of block length N with relative distance $\geq \delta$ and rate

$$R \;\geq\; 1 - H(\delta) - \varepsilon.$$

Proof idea: the probabilistic method — pick $M = 2^{RN}$ codewords independently and uniformly at random, and show that with positive probability no two are close.

Sketch: concentration, then a union bound

- one pair: $d(c_i, c_j) \sim \mathrm{Bin}(N, \frac{1}{2})$, concentrated at $N/2$
- Chernoff / ball volume: $\Pr\left[d(c_i, c_j) < \delta N\right] \le 2^{-(1-H(\delta))N}$
- union bound over $< M^2/2$ pairs: good code exists once $M^2 \ll 2^{(1-H(\delta))N}$
- that gives rate $\frac{1-H(\delta)}{2}$; expurgation (delete one endpoint of every bad pair) doubles it back to $1 - H(\delta)$

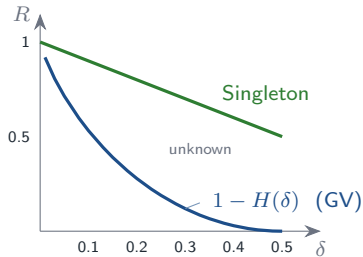


The rate–distance question

Fix the relative distance $\delta = d/N$.
How large can the rate R be?

- **Singleton:** $R \le 1 - \delta + \frac{1}{N}$
- **Gilbert–Varshamov:**
  $R \ge 1 - H(\delta)$ is achievable

H: binary entropy. For binary codes, closing this gap is a major open problem.



Linear codes

Definition

$\mathcal{C} \subseteq \mathbb{F}^N$ is a **linear code** if it is a linear subspace.

- dimension n = message length; rate $R = n/N$
- distance $= \min_{0 \neq w \in \mathcal{C}} \mathrm{wt}(w)$, the minimum *weight*

why the two notions of distance agree: $c_1 - c_2 \in \mathcal{C}$ and $d(c_1, c_2) = \mathrm{wt}(c_1 - c_2)$

a subspace is given by a generator matrix $A \in \mathbb{F}^{N \times n}$: $\mathcal{C} = \{Ax : x \in \mathbb{F}^n\}$

Random linear codes are good too

Theorem

A uniformly random $A \in \mathbb{F}^{N \times n}$ generates a code of relative distance $\geq \delta$ with high probability, at rate up to the Gilbert–Varshamov bound.

Same recipe:

- for a fixed $x \neq 0$, Ax is uniform in $\mathbb{F}^N$: $\mathbb{E}\,\mathrm{wt}(Ax) = \frac{|\mathbb{F}|-1}{|\mathbb{F}|}N$, and $\Pr[\mathrm{wt}(Ax) < \delta N] \leq e^{-cN}$
- union bound over the $|\mathbb{F}|^n - 1$ nonzero *messages* — not over pairs

storage: $Nn \log |\mathbb{F}|$ bits for a generator matrix, versus $|\mathbb{F}|^n \cdot N \log |\mathbb{F}|$ for a general code

What is still missing: decoding

- ✓ random (linear) codes have excellent rate–distance trade-offs
- ✓ a linear code is cheap to store
- × decoding a random linear code appears hard:
  worst case NP-hard (Berlekamp–McEliece–van Tilborg), average case = *learning parity with noise*

We need structure — enough structure to decode, without losing distance.

Outline

Codes, distance, decoding

How good can codes be?

Reed–Solomon codes

Unique decoding: Welch–Berlekamp

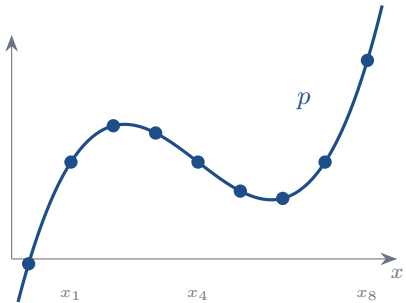
List decoding

Reed–Solomon codes

Fix a finite field $\mathbb{F}$ and distinct evaluation points $x_1, \ldots, x_N \in \mathbb{F}$.

$$\mathrm{RS}(N, n) \;=\; \Big\{ \big(p(x_1), \ldots, p(x_N)\big) \;:\; p \in \mathbb{F}[x],\ \deg p < n \Big\}$$

- message $=$ coefficients $(a_0, \ldots, a_{n-1})$
- codeword $=$ the table of values of p
- needs $|\mathbb{F}| \geq N$: a *large* alphabet



a degree-$< n$ polynomial, sampled at N points

Why Reed–Solomon is a good code

- **linear:** $p_1 + p_2$ and λp again have degree $< n$
- **dimension n:** a polynomial is determined by its n coefficients
- **distance $d = N - n + 1$:** a nonzero p with $\deg p < n$ has at most $n - 1$ roots, so at least $N - n + 1$ of its values are nonzero

$$R = \frac{n}{N}, \quad \delta = 1 - R + \frac{1}{N}$$ — Reed–Solomon meets the Singleton bound

e.g. $n = 0.1N$: rate 0.1, distance $\approx 0.9N$, so up to $0.45N$ worst-case errors — *if* we can decode.

Outline

Codes, distance, decoding

How good can codes be?

Reed–Solomon codes

Unique decoding: Welch–Berlekamp

List decoding

The decoding problem

Given

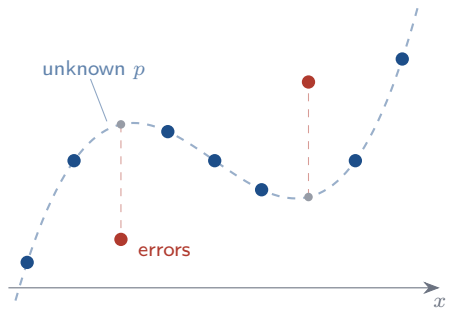
values $\tilde{y}_1, \ldots, \tilde{y}_N \in \mathbb{F}$ with

$$\tilde{y}_i = p(x_i) \quad \text{for all but} \leq t \text{ indices},$$

for some unknown p of degree $< n$.

Find p.

- for $t < d/2$ the answer is unique — the issue is finding it
- n clean values determine p, but which ones are clean? $\binom{N}{n}$ guesses

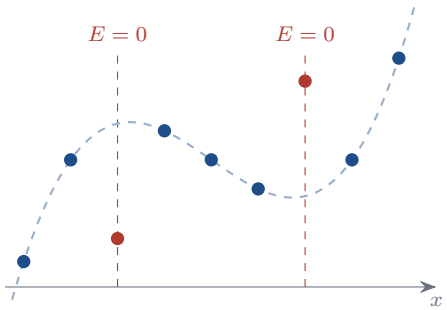


the received word: clean values, corrupted ones

Key idea: the error locator polynomial

Let $B \subseteq [N]$ be the set of corrupted positions, $|B| \le t$, and

$$E(x) \;:=\; \prod_{i \in B} (x - x_i), \qquad \deg E \le t.$$



E vanishes exactly on the corrupted positions

Then for every $i \in [N]$

$$\tilde{y}_i \, E(x_i) \;=\; p(x_i) \, E(x_i)$$

- $i \notin B$: $\tilde{y}_i = p(x_i)$
- $i \in B$: $E(x_i) = 0$, both sides vanish

Write $W := p\,E$ $(\deg W \le t + n - 1)$: the identity reads $W(x_i) = \tilde{y}_i \, E(x_i)$ — linear in (W, E).

Welch–Berlekamp: solve for the pair, not for p

The linear system

Find polynomials $(U, V) \neq (0, 0)$ with

$$\deg V \leq t, \qquad \deg U \leq t + n - 1, \qquad U(x_i) = \tilde{y}_i \, V(x_i) \quad \text{for all } i \in [N].$$

- unknowns: the $(t+1) + (t+n)$ coefficients of V and U
- equations: N, homogeneous and linear in those coefficients
- solvable by Gaussian elimination — $O(N^3)$ field operations

Note what is *not* asked: we never guess B, E or p.

The system is feasible

The pair

$$(V, U) \;=\; (E, pE)$$

is a solution:

- $\deg E \le t$ ✓
- $\deg pE \le (n-1) + t$ ✓
- $pE(x_i) = \tilde{y}_i E(x_i)$ ✓
- $E \neq 0$ ✓

But beware

Gaussian elimination returns *some* nonzero solution — not necessarily (E, pE).

The zeros of the V we get need not be the error positions.

Remarkably, every solution is good enough.

Every solution yields p

Let $(U, V) \neq (0, 0)$ be any solution, and set $D := U - pV$.

1. $\deg D \leq \max\{\, t + n - 1,\ (n-1) + t \,\} = t + n - 1$
2. at each clean position: $D(x_i) = \tilde{y}_i V(x_i) - p(x_i)V(x_i) = 0$ — at least $N - t$ roots
3. if $N - t > t + n - 1$ then D has more roots than its degree: $D \equiv 0$
4. $V \not\equiv 0$: else U vanishes at all N points while $\deg U \leq t + n - 1 < N$

$$U = pV \text{ with } V \neq 0 \implies p = \frac{U}{V} \quad \text{(an exact polynomial division)}$$

The radius is exactly half the distance

The condition for the proof was

$$N - t \;>\; t + n - 1 \qquad \Longleftrightarrow \qquad t \;\le\; \left\lfloor \frac{N-n}{2} \right\rfloor \;=\; \left\lfloor \frac{d-1}{2} \right\rfloor .$$



Theorem (Welch–Berlekamp)

$\mathrm{RS}(N, n)$ can be decoded from any $t \le \frac{N-n}{2}$ worst-case errors in time $O(N^3)$ — the full unique-decoding radius $\lfloor (d-1)/2 \rfloor$.

With structured linear algebra (Berlekamp–Massey, fast interpolation) this drops to $\tilde{O}(N)$.

Outline

Codes, distance, decoding

How good can codes be?

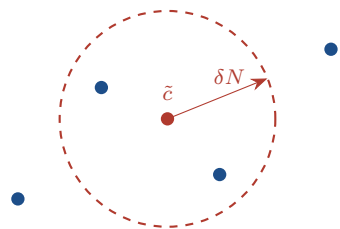
Reed–Solomon codes

Unique decoding: Welch–Berlekamp

List decoding

Beyond half the distance

- unique decoding stops at $d/2$ — but only because of rare *midpoint* words
- for a typical received word, one codeword is still far closer than all others
- relaxation: output *all* codewords within distance δN, and hope the list is short



a short list instead of a unique answer

List decoding

Definition

$\mathcal{C}$ is (δ, L)-**list-decodable** if every word $w \in \Sigma^N$ has at most L codewords within distance δN.

Useful when $L = \mathrm{poly}(N)$ *and* the list is computable in polynomial time.

Theorem (Sudan '97)

$\mathrm{RS}(N, n)$ is efficiently (δ, L)-list-decodable for

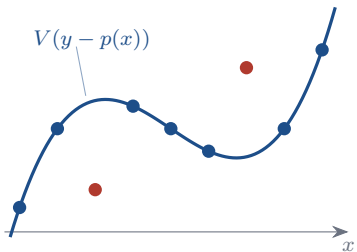
$$\delta \;=\; 1 - 2\sqrt{n/N} \;=\; 1 - 2\sqrt{R}, \qquad L \;\le\; \sqrt{N/n} \;=\; 1/\sqrt{R}.$$

$R = 0.01$: unique decoding $0.495N$ errors; Sudan $0.8N$; Guruswami–Sudan $0.9N$.

Dictionary: polynomials are curves

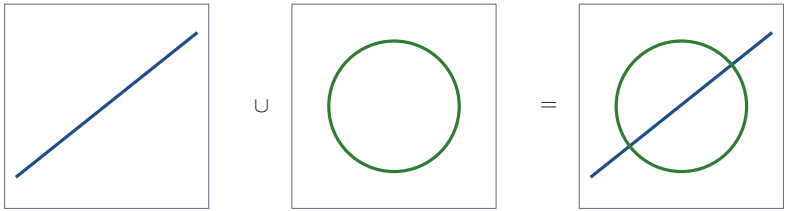
For $q \in \mathbb{F}[x, y]$ let
$V(q) = \{(x, y) : q(x, y) = 0\}$.

- a message polynomial p $\rightsquigarrow$ the curve $V(y - p(x))$, i.e. its graph
- the received word $\rightsquigarrow$ the N points $(x_i, \tilde{y}_i)$
- "p agrees with $\tilde{y}$ in $\geq (1 - \delta)N$ places" $\rightsquigarrow$ "the curve passes through $\geq (1 - \delta)N$ of the points"



decoding = find the curves through many points

Products of polynomials = unions of curves



$q(x, y) = 0$ at a point $\iff$ some factor of q vanishes there

Consequence: a *single* low-degree q can carry many graphs at once — one factor $y - p(x)$ per candidate message.

Bézout: curves without a common factor meet rarely

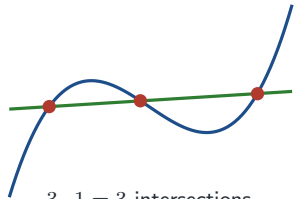
Theorem (Bézout)

If q_1, q_2 have no common factor, then

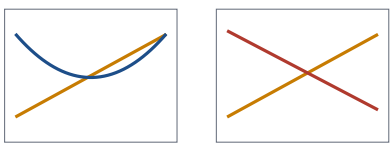
$$|V(q_1) \cap V(q_2)| \;\leq\; \deg q_1 \cdot \deg q_2.$$

Contrapositive — the one we use

too many common points $\Longrightarrow$ a common factor



$3 \cdot 1 = 3$ intersections

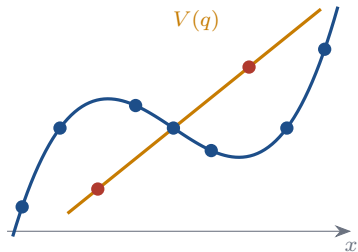


$V(q_1)$ $V(q_2)$

common component: infinitely many

The plan

1. fit *one* low-degree curve $V(q)$ through all N received points
2. a p agreeing with $\tilde{y}$ often $\Rightarrow$ its graph meets $V(q)$ often $\Rightarrow$ $y - p(x)$ is a factor of q
3. factor q; the factors $y - p(x)$ are the candidate messages



$q = (y - p(x))\,(y - g(x))$: one curve through *every* point

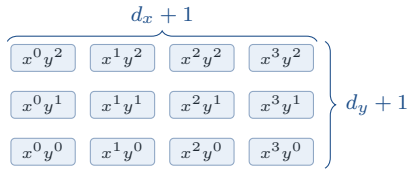
Step 1: fitting q is linear algebra

Look for $q \neq 0$ with $\deg_x q \leq d_x$, $\deg_y q \leq d_y$:

$$q(x, y) = \sum_{a \leq d_x} \sum_{b \leq d_y} c_{ab}\, x^a y^b.$$

- the N conditions $q(x_i, \tilde{y}_i) = 0$ are homogeneous linear in the c_{ab}
- more unknowns than equations $\Rightarrow$ a nonzero solution exists:

$$(d_x + 1)(d_y + 1) > N$$



the monomials of q — the unknowns

Step 2: much agreement forces a factor

Let p ($\deg p < n$) agree with $\tilde{y}$ on a set A, $|A| \geq N - t$, and put

$$r(x) \;:=\; q\big(x, p(x)\big).$$

- $\deg r \;\leq\; d_x + (n-1)\, d_y$
- for $i \in A$: $\; r(x_i) = q(x_i, \tilde{y}_i) = 0$ — at least $N - t$ roots
- so if $\; N - t > d_x + (n-1)d_y \;$ then $\; r \equiv 0$, i.e. $q(x, p(x))$ is the zero polynomial

$$q(x, p(x)) \equiv 0 \quad \Longrightarrow \quad \big(y - p(x)\big) \mid q(x, y)$$

the same "many intersections $\Rightarrow$ common factor" as Bézout, now with the sharper degree bound

Choosing the two degrees

Two constraints pull in opposite directions:

$$\underbrace{(d_x+1)(d_y+1) > N}_{q \text{ exists}} \qquad \text{and} \qquad \underbrace{N - t > d_x + (n-1)d_y}_{p \text{ is a factor}}.$$

The balanced choice

$$d_x = \sqrt{Nn}, \qquad d_y = \sqrt{N/n} \qquad (\text{so } d_x d_y = N)$$

- $d_x + (n-1)d_y < 2\sqrt{Nn}$, so $t < N - 2\sqrt{Nn}$ suffices: $\delta < 1 - 2\sqrt{n/N}$
- **list size:** distinct candidates give distinct factors $y - p(x)$, and $\deg_y q \le d_y$, so $L \le \sqrt{N/n}$

Summary and outlook

The list decoder

1. solve a linear system for q with $\deg_x q \le \sqrt{Nn}$, $\deg_y q \le \sqrt{N/n}$
2. factor q (efficient over finite fields)
3. output the p with $y - p(x) \mid q$ that agree with $\tilde{y}$ in $\ge (1-\delta)N$ places

- plain Bézout ($d_x = d_y = \sqrt{N}$) already gives $\delta = 1 - 2(n-1)/\sqrt{N}$
- Guruswami–Sudan: ask q to vanish with *multiplicity* at each point $\Rightarrow \delta = 1 - \sqrt{R}$
- folded Reed–Solomon codes reach the capacity $\delta = 1 - R$
- list decoding is a tool far beyond channels: hardness amplification, pseudorandomness, cryptography