Reed-Solomon Codes

The Greatest Code of Them All

Anakin

Section 1

Introduction

Idea

- How many roots does a degree d polynomial have?
  - Fundamental Theorem of Algebra say $\leq d$ roots
- Idea: Encode messages as evaluations of polynomials
- Few roots $\implies$ good distance

Notation:

- $\mathbb{F}_q$ = "Integers mod q" where q is a prime power (Finite Field)
- $\mathbb{F}_q[x]$ = Polynomials with coefficients in $\mathbb{F}_q$

Definition (Reed-Solomon Codes [RS60])

Let $q \geq n \geq k$. Let $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$ be distinct *evaluation points*. The *Reed-Solomon Code* of dimension k with alphabet $\mathbb{F}_q$ and evaluation points $\vec{\alpha} = [\alpha_1, \ldots, \alpha_n]$ is

$$\mathrm{RS}_q(\vec{\alpha}, n, k) = \{\, [f(\alpha_1), \ldots, f(\alpha_n)] \mid f \in \mathbb{F}_q[x],\ \deg(f) \leq k - 1 \,\}$$

Encoding

Definition (Reed-Solomon Codes [RS60])

Let $q \geq n \geq k$. Let $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$ be distinct *evaluation points*. The *Reed-Solomon Code* of dimension k with alphabet $\mathbb{F}_q$ and evaluation points $\vec{\alpha} = [\alpha_1, \ldots, \alpha_n]$ is

$$\mathrm{RS}_q(\vec{\alpha}, n, k) = \{\, [f(\alpha_1), \ldots, f(\alpha_n)] \mid f \in \mathbb{F}_q[x],\ \deg(f) \leq k-1 \,\}$$

Say we want to encode a message $\vec{m} = [m_0, m_1, \ldots, m_{k-1}]$, $m_i \in \mathbb{F}_q$

Let $f_{\vec{m}}(x) \coloneqq \sum_{i=0}^{k-1} m_i x^i$

$$\mathrm{ENC}(m_0, \ldots, m_{k-1}) = [f_{\vec{m}}(\alpha_1), \ldots, f_{\vec{m}}(\alpha_n)]$$

Linearity

Definition (Reed-Solomon Codes [RS60])

$\mathrm{RS}_q(\vec{\alpha}, n, k) = \{ [f(\alpha_1), \ldots, f(\alpha_n)] \mid f \in \mathbb{F}_q[x],\ \deg(f) \leq k - 1 \}$

- $\mathrm{RS}_q(\vec{\alpha}, n, k)$ is a *linear* code
  - Polynomials of degree $\leq k - 1$ in $\mathbb{F}_q[x]$ are a k-dimensional vector space
- If you recall from Hassam's meeting on linear codes, linear codes have generator matrices

$$G = \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{k-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{k-1} \\ \vdots & \vdots & \alpha_2^2 & \cdots & \vdots \\ 1 & \alpha_n & \alpha_n^2 & \cdots & \alpha_n^{k-1} \end{bmatrix}$$

Example

Let $q \geq n \geq k$ be $7 \geq 4 \geq 3$ respectively. Let $\vec{\alpha} = [\alpha_1, \dots, \alpha_n] = [1, 2, 4, 6]$.

$$\mathrm{ENC}(m_0, \dots, m_{k-1}) = [f_{\vec{m}}(\alpha_1), \dots, f_{\vec{m}}(\alpha_n)]$$

Lets encode $[m_0, m_1, m_2] = [1, 3, 5]$.

$$f_{\vec{m}}(x) \coloneqq \sum_{i=0}^{k-1} m_i x^i = ???$$

Example

Let $q \geq n \geq k$ be $7 \geq 4 \geq 3$ respectively. Let $\vec{\alpha} = [\alpha_1, \ldots, \alpha_n] = [1, 2, 4, 6]$.

$$\mathrm{ENC}(m_0, \ldots, m_{k-1}) = [f_{\vec{m}}(\alpha_1), \ldots, f_{\vec{m}}(\alpha_n)]$$

Lets encode $[m_0, m_1, m_2] = [1, 3, 5]$.

$$f_{\vec{m}}(x) := \sum_{i=0}^{k-1} m_i x^i = 1 + 3x + 5x^2 \in \mathbb{F}_7[x]$$

$$[f_{\vec{m}}(\alpha_1), f_{\vec{m}}(\alpha_2), f_{\vec{m}}(\alpha_3), f_{\vec{m}}(\alpha_4)] = [2, 6, 2, 3]$$

How do we *decode*?

The Original Decoding Algorithm

Say we want to recover $\vec{m}$ from $[f(\alpha_1), \ldots, f(\alpha_n)]$. Here is the original algorithm from Reed and Solomon's paper:

1. $\deg(f) = k - 1$ so choose any k of the evaluations $f(\alpha_i)$
2. *Interpolate* these points and find the polynomial $f_{\vec{m}'}(x) = \sum_{i=0}^{k-1} m'_i x^i$ defined by these points
3. Do this for all $\binom{n}{k}$ choices of evaluations, do majority voting to pick out the right coefficients m_i

Lagrange Interpolation

- *Note:* If two polynomials of degree $\leq k-1$ agree on k points, they must be the same polynomial
- Let $f(x)$ be some polynomial of degree $\leq k-1$, $\alpha_1, \ldots, \alpha_k$ distinct points in $\mathbb{F}_q$

$$L(x) \coloneqq \sum_{i=1}^{k} f(\alpha_i) \left(\prod_{\substack{j=1 \\ i \neq j}}^{k} \frac{x - \alpha_j}{\alpha_i - \alpha_j} \right) = \text{ the } \textit{Lagrange Interpolating polynomial}$$

Claim: $L(x) = f(x)$ for all x.

Lagrange Interpolation

Let $f(x)$ be some polynomial of degree $\leq k-1$, $\alpha_1, \ldots, \alpha_k$ distinct points in $\mathbb{F}_q$

$$L(x) \coloneqq \sum_{i=1}^{k} f(\alpha_i) \left(\prod_{\substack{j=1 \\ i \neq j}}^{k} \frac{x-\alpha_j}{\alpha_i - \alpha_j} \right) = \text{ the } \textit{Lagrange Interpolating polynomial}$$

$$\prod_{\substack{j=1 \\ i \neq j}}^{k} \frac{x-\alpha_j}{\alpha_i - \alpha_j} = \begin{cases} 1 & \text{if } x = \alpha_i \\ 0 & \text{if } x \neq \alpha_i \text{ (so } x = \alpha_j \text{ for some } i \neq j) \end{cases}$$

Lagrange Interpolation

Let $f(x)$ be some polynomial of degree $\leq k-1$, $\alpha_1, \ldots, \alpha_k$ distinct points in $\mathbb{F}_q$

$$L(x) := \sum_{i=1}^{k} f(\alpha_i) \left(\prod_{\substack{j=1 \\ i \neq j}}^{k} \frac{x - \alpha_j}{\alpha_i - \alpha_j} \right) = \text{ the } \textit{Lagrange Interpolating polynomial}$$

$$f(\alpha_i) \cdot \prod_{\substack{j=1 \\ i \neq j}}^{k} \frac{x - \alpha_j}{\alpha_i - \alpha_j} = \begin{cases} f(\alpha_i) & \text{if } x = \alpha_i \\ 0 & \text{if } x \neq \alpha_i \text{ (so } x = \alpha_j \text{ for some } i \neq j) \end{cases}$$

Lagrange Interpolation

Let $f(x)$ be some polynomial of degree $\leq k-1$, $\alpha_1, \ldots, \alpha_k$ distinct points in $\mathbb{F}_q$

$$L(x) \coloneqq \sum_{i=1}^{k} f(\alpha_i) \left(\prod_{\substack{j=1 \\ i \neq j}}^{k} \frac{x - \alpha_j}{\alpha_i - \alpha_j} \right) = \text{ the } \textit{Lagrange Interpolating polynomial}$$

$$\sum_{i=1}^{k} f(\alpha_i) \left(\prod_{\substack{j=1 \\ i \neq j}}^{k} \frac{x - \alpha_j}{\alpha_i - \alpha_j} \right) = f(\alpha_i) \text{ if } x = \alpha_i \text{ for some } i$$

$\implies L(x) = f(x)$ on k points $\implies L(x) = f(x)$ on all points

History

- 1960: Reed and Solomon publish their original paper [RS60]
- 1968/1969: Berlekamp and Massey improve on the decoding algorithm [Ber68, Mas69]
- 1986: Berlekamp and Welch design an even faster decoding algorithm [WB86]
- 1996 and beyond: List Decoding methods become more prevalent

Applications

QR Codes

CDs were the first mass produced item to use Reed-Solomon Codes (combined with techniques for burst errors from last meeting!)

Questions?

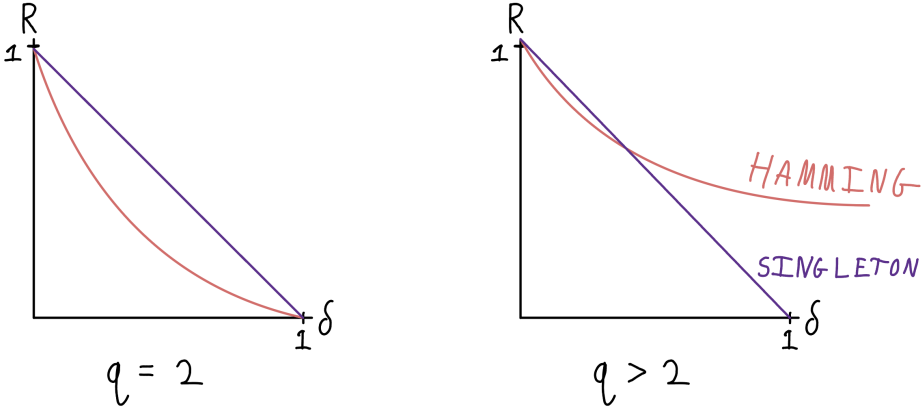
Section 2

The Singleton Bound

Theorem (Singleton Bound [Sin64])

If C is is code over an alphabet Σ of size q encoding messages of length k into codes of length n with distance d, then $k \leq n - d + 1$

Recall $R = \frac{k}{n}$ is the *rate* of C



Proof of the Singleton Bound

Theorem (Singleton Bound [Sin64])

If C is is code over an alphabet Σ of size q encoding messages of length k into codes of length n with distance d, then $k \leq n - d + 1$

- Let $c = (\overbrace{c_1, \ldots, c_{n-d+1}}^{\phi(c)}, \overbrace{c_{n-d+2}, \ldots, c_n}^{\text{discard}}) \in C$
- Let $\widetilde{C} = \{ \phi(c) \mid c \in C \} \subseteq \Sigma^{n-d+1}$
- *Claim:* $|C| = |\widetilde{C}|$
  - If not, then $\phi(c) = \phi(c')$ for some $c \neq c' \in C$. Thus $\Delta(c, c') \leq d - 1$. Contradicts fact that C has distance d!
- Thus $|C| = |\widetilde{C}| \leq q^{n-d+1}$
- $\implies k := \log_q |C| \leq n - d + 1$

Theorem

Reed-Solomon codes meet the Singleton Bound! The distance of $\mathrm{RS}_q(\vec{\alpha}, k, n)$ is $d = n - k + 1$.

- The intuition for this is that the polynomials can have at most $k-1$ zeroes so at most $k-1$ of the $f(\alpha_i)$'s are 0.
- Distance $n - k - 1$ means we can correct $\leq \left\lfloor \frac{n-k}{2} \right\rfloor$ errors
- We call Linear $(n, k, d)_q$ codes with distance $d = n - k + 1$ *Maximum Distance Seperable* codes.

Conjecture (The MDS Conjecture [Seg55])

If $k \leq q$ then a linear MDS code has $n \leq q + 1$ unless $q = 2^h$ and $k = 3$ or $k = q - 1$ in which case $n \leq q + 2$.

Questions?

Section 3

Berlekamp-Welch

A Faster Decoding Algorithm

Given $(c_1, \ldots, c_n \in \mathrm{RS}_q(\vec{\alpha}, n, k)$ with $e \leq \left\lfloor \frac{n-k}{2} \right\rfloor$ errors, we want to find $f \in \mathbb{F}_q[x]$ such that $\deg(f) \leq k$ and $f(\alpha_i) \neq c_i$ at most e times.

Here is the idea behind the faster Berlekamp-Welch algorithm for decoding Reed-Solomon codes:

- Let $E(x) \coloneqq \prod_{i:\, c_i \neq f(\alpha_i)} (x - \alpha_i)$ be the *Error Locator Polynomial*
- For all $1 \leq i \leq n$, we have that $c_i \cdot E(\alpha_i) = f(\alpha_i) \cdot E(\alpha_i)$
  - If $c_i = f(\alpha_i)$ then this is obvious
  - If $c_i \neq f(\alpha_i)$ then this is true because $E(\alpha_i) = 0$
- The algorithm will find $E(x)$ and $Q(x) \coloneqq f(x) \cdot E(x)$

These Polynomials Exist

Lemma

Suppose there was some degree $\leq k-1$ polynomial $f_{\vec{m}}(x) = \sum_{i=0}^{k-1} m_i x^i$ such that $\Delta(m, c) \leq e$. Then there exist polynomials $E(x)$ *monic* of degree $\leq e$ and $Q(x)$ of degree $\leq e+k-1$ such that

$$\text{for all } 1 \leq i \leq n,\ c_i \cdot E(\alpha_i) = Q(\alpha_i)$$

- $E(x) = \left(\prod_{i:\ c_i \neq f(\alpha_i)} (x - \alpha_i) \right) * x^{e - \Delta(m,c)}$
- $Q(x) = E(x) \cdot f(x)$

System of Equations

Ok but how do we find $E(x)$ and $Q(x)$?

$$E(x) \coloneqq \sum_{j=0}^{e} e_j x^j \qquad\qquad Q(x) \coloneqq \sum_{j=0}^{e+k-1} q_j x^j$$

Since $c_i \cdot E(\alpha_i) = Q(\alpha_i)$, we have $c_i \cdot E(\alpha_i) - Q(\alpha_i) = 0$ for all i. This gives n linear equations, one for each α_i:

$$c_i \sum_{j=0}^{e} e_j \alpha_i^j - \sum_{j=0}^{e+k-1} q_j \alpha_i^j = 0$$

We have n linear equations, and $2e + k$ variables. The *Lemma* tells us that if there are not too many errors, some solution exists!

BERLEKAMP-WELCH($c = [c_1, \dots, c_n]$):

Find polynomials $E(x), Q(x) \in \mathbb{F}_q[x]$ such that

  $E(x)$ is monic of degree e

  $Q(x)$ is of degree $\le e + k - 1$

  For all $1 \le i \le n$:

$$c_i \cdot E(\alpha_i) = Q(\alpha_i)$$

If no solution: return NONE

$\widetilde{f_{\vec{m}}} \leftarrow Q(x)/E(x)$

$\tilde{c} \leftarrow \mathrm{ENC}(m_0, \dots, m_{k-1})$

If $\Delta(\tilde{c}, c) > e$, return NONE

Return $\widetilde{f_{\vec{m}}}$

1. Gaussian Elimination takes $O(n^3)$ time
2. Polynomial division will take $O(n^3)$ time
3. Computing $\tilde{c}$ will take $O(nk^2) \le O(n^3)$ time
4. Computing $\Delta(\tilde{c}, c)$ takes $O(n)$ time

Questions?

So long and thanks for all the fish!

— DOUGLAS ADAMS (1979)

Bibliography I

E. Berlekamp.

Nonbinary bch decoding (abstr.).

IEEE Transactions on Information Theory, 14(2):242–242, 1968.

J. Massey.

Shift-register synthesis and bch decoding.

IEEE Transactions on Information Theory, 15(1):122–127, 1969.

I. S. Reed and G. Solomon.

Polynomial codes over certain finite fields.

Journal of the Society for Industrial and Applied Mathematics, 8(2):300–304, 1960.

Beniamino Segre.

Curve razionali normali ek-archi negli spazi finiti.

Annali di Matematica Pura ed Applicata, 39(1):357–379, December 1955.

Bibliography II

R. Singleton.

Maximum distanceq-nary codes.

IEEE Transactions on Information Theory, 10(2):116–118, 1964.

Lloyd R. Welch and Elwyn R. Berlekamp.

Error correction for algebraic block codes, 12 1986.

US Patent 4633470.

Mary Wootters.

Algebraic coding theory, 2021.

Neal Zierler.

An introduction to algebraic and combinatorial coding theory (ian f. blake and ronald c. mullin).

SIAM Review, 20(3):607–608, 1978.