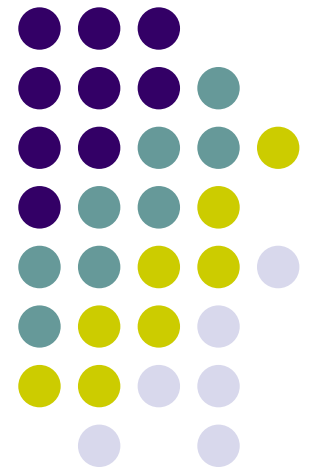


CS257

Introduction to Nanocomputing

Codes and Finite Fields

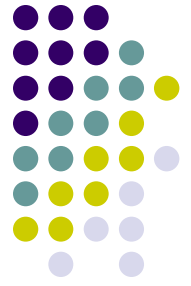
John E Savage





Lecture Outline

- Motivation
- Error Correcting Codes
- Reed Solomon Codes
- Spielman's [approach](#)



Efficient Reliable Circuits

- **The goal:** To reduce the redundancy of an unreliable circuit simulating a reliable one.
- **The approach:** To replace the repetition code with a more efficient one.



Building Reliable Circuits

- Prevent gate failures from making circuit failure rates prohibitively high.
- Use error correcting codes to detect and correct circuit failures.



Error Correcting Codes

- An error-correcting code is a set of n -tuples over an alphabet Σ , called **codewords**.
- The **distance** between two codewords is the number of places in which they differ.
- The **minimum distance** of a code is the minimum over all pairs of codewords of the distance between them.



$(n, k, d)_q$ Block Codes

- An $(n, k, d)_q$ block code.
 - **Message length** = k
 - **Block length** n
 - **Rate** $R = k/n$
 - **Minimum distance** d
 - **Alphabet size** = q
- Shannon showed that, as k increase, R need not go to 0 to accommodate an error rate $< .5$
- It is not known if this holds for computation.



Hamming Code

- Encode $\mathbf{b} = (b_0, b_1, b_2, b_3)$ as $\mathbf{b}G$ where

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- G is the **generator matrix**.
- This is a $(7,4,3)_2$ code. Why is $d = 3$?
 - Compare \mathbf{b}_1G and \mathbf{b}_2G where $\mathbf{b}_1 \neq \mathbf{b}_2$.
 - Note that $\mathbf{b}_1G \oplus \mathbf{b}_2G$ (term-by-term XOR) is equivalent to \mathbf{b}_3G where $\mathbf{b}_3 = \mathbf{b}_1 \oplus \mathbf{b}_2$.



Generalized Hamming Code

- Let $n = 2^k - 1$. The $(n, k, 3)_2$ Hamming code has the following generator matrix.

$$G = \begin{bmatrix} I_{k \times k} & B_{k \times n-k} \end{bmatrix}$$

- Here $B_{k \times n-k}$ contains all k -tuples except for 0^{n-k} and the weight 1 k -tuples.



Decoding Hamming Codes

- Let $n = 2^k - 1$. Form $n \times k$ matrix H .

$$H = \begin{bmatrix} B_{k \times n-k} \\ I_{n-k \times n-k} \end{bmatrix}$$

- If \mathbf{w} is a Hamming codeword, $\mathbf{w}H = \mathbf{0}$.
- If $\mathbf{w} \oplus \mathbf{e}$ is received, $\mathbf{s} = (\mathbf{w} \oplus \mathbf{e})H = \mathbf{e}H$. Since all single errors can be corrected ($|\mathbf{e}| = 1$), each **syndrome** \mathbf{s} is associated with a unique row of H !



Linear Block Codes

- Generalization of Hamming Codes
- In a linear block code, the vector sum of two codewords is another codeword.
- Linear codes can be defined by generator matrices.
 - A basis exists for this linear space
 - A codeword is linear combination of basis vectors.



Binary Error Correcting Codes

- Let addition over Σ be \oplus (Exclusive OR)
- The **Hamming distance** $d(\mathbf{c}, \mathbf{c}')$ between two binary codewords \mathbf{c} , \mathbf{c}' is the weight (number of 1s in) of their component-wise sum \oplus .

$$(0, 1, 1, 0, 0, 1) \oplus (1, 1, 0, 1, 0, 1) = (1, 0, 1, 1, 0, 0)$$

- $d(\mathbf{c}, \mathbf{c}') = |(1, 0, 1, 1, 0, 0)| = 3$.



Non-Binary Codes

- Codewords defined over non-binary Σ .
 - Generally $\Sigma = F$, a finite field.
 - All finite fields have $|F| = p^m$ for prime p and integer m . They are called Galois fields $GF(p^m)$.
 - Fields have addition (+) and multiplication (*) operators, constants 0 and 1. Usual associative and distributive laws hold.
 - Elements of $GF(q)$ are $\{0, 1, \alpha, \alpha^2, \dots, \alpha^{q-2}\}$, $q=p^m$
- **Linear codes** are codes in which the vector sum of two codewords is another codeword.



Generating Linear Codewords

- Codewords are linear combinations of the rows of a $k \times n$ matrix

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

- A linear combination results from pre-multiplication of G by a binary vector $\mathbf{u} = (u_0, u_1, u_2)$
 - $(1, 1, 0)G = (1, 1, 0, 1, 0)$.
 - Codeword $\mathbf{c} = (u_0, u_1, u_2, c_1, c_2)$ where u_i is an **information bit** and c_j is a **check bit**



More on Linear Codewords

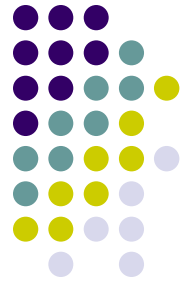
- Assume without loss of generality that rows of generator matrix are linearly independent.
- Given input $\mathbf{u} \in F^k$, its codeword is $\mathbf{c} = \mathbf{uG}$.
- A $k \times n$ generator matrix can be put into **standard form** by elementary row operations and column permutations, $G = [I_k, A]$, where I_k is the $k \times k$ identity matrix and A is a $k \times (n-k)$ matrix over F .



The Parity Check Matrix

- The **parity check matrix** $H = \begin{bmatrix} -A \\ I_{n-k} \end{bmatrix}$ where I_{n-k} is the $(n-k) \times (n-k)$ identity matrix.
- Every codeword \mathbf{c} generated by G is in the **null space** of H , that is, $\mathbf{c}H = \mathbf{0}$.
 - This follows because for some \mathbf{u} , $\mathbf{c} = \mathbf{u}G$ and $GH = [I_k(-A) + AI_{n-k}] = \mathbf{0} = [0_k]$ where 0_k is the $k \times k$ zero matrix.

The Minimum Distance of a Linear Code



- The **Hamming distance** $d(\mathbf{c}_1, \mathbf{c}_2)$ between two linear codewords \mathbf{c}_1 and \mathbf{c}_2 is the number of non-zero components in their term-by-term difference $\mathbf{c}_1 - \mathbf{c}_2$, that is, $d(\mathbf{c}_1, \mathbf{c}_2) = |\mathbf{c}_1 - \mathbf{c}_2|$.
- Because the difference between codewords in a linear code is another codeword, the **minimum distance** d is the weight of the smallest weight codeword.

Minimum Distance (Projection) Bound

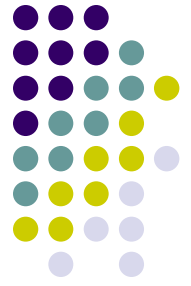


- Distance bound for $(n,k,d)_q$ codes: **$d \leq n-k+1$**
 - Project the q^k codewords onto first $k-1$ positions.
 - By pigeon-hole principle, at least two codewords have these k positions in common.
 - Thus, the minimum distance $d \leq n-k+1$.



Correcting Errors

- If a codeword \mathbf{c} is sent over a noisy channel and e errors occur, $e \leq (d-1)/2$, the resulting word $\mathbf{r} = \mathbf{c} + \mathbf{e}$ is closer (has fewer differences from) to the transmitted word than to any other codeword.
 - For $\mathbf{c}' \neq \mathbf{c}$, $d(\mathbf{c}', \mathbf{c}) = |\mathbf{c}' - \mathbf{c}| = |\mathbf{c}' - \mathbf{r} + \mathbf{r} - \mathbf{c}| \leq |\mathbf{c}' - \mathbf{r}| + |\mathbf{r} - \mathbf{c}|$ but $|\mathbf{c}' - \mathbf{c}| \geq d$ and $|\mathbf{r} - \mathbf{c}| = e$. Thus, $|\mathbf{c}' - \mathbf{r}| \geq (d+1)/2$ and \mathbf{r} is closer to \mathbf{c} than to any other codeword.
- Errors stat. independent with prob. p
- $P(e \text{ errors}) = \binom{n}{e} p^e (1 - p)^{n-e}$
- Minimizing e minimizes prob of error



Decoding a Linear Code

- Given r , find closest codeword c' , i.e. $D(r) = c'$.
 - Can decoding errors occur?
- Equivalently, given received word r compute the **syndrome** $s = rH = (c+e)H = eH$.
 - The syndrome is a function only of the errors
 - Possible that $r = c' + e'$ where $|e'| \leq |e|$.
- Given r find smallest weight e' satisfying s . Add to r .

$(n,k,d)_q$ Reed Solomon Codes



- To encode **message** $(a_0, a_1, \dots, a_{n-1})$, a_i in $GF(q)$, evaluate $s(x) = a_0 + a_1x + \dots + a_{n-1}x^{n-1}$ for all x in $GF(q)$
- Codeword associated with $(a_0, a_1, \dots, a_{n-1})$ is $\mathbf{s} = (r(0), r(1), r(\alpha), r(\alpha^2), \dots, r(\alpha^{q-2}))$
 - Given y in $GF(q)$, the n such that $y = \alpha^n$ is the **discrete log**. It arises in cryptography.



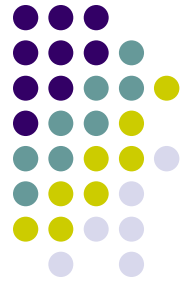
Fields $(F, +, \times, 0, 1)$

- F is a countable set, $+$ and \times are associative “addition” and “multiplication” operators
- 0 & 1 are identity under addition and multiplication respectively.
 - F is commutative and associative under $+$ and \times .
 - \times distributes over $+$
 - Additive inverse exists for each element
 - Multiplicative inverse exists for $F - \{0\}$.



Finite Fields (Galois Fields)

- All finite fields have p^n elements for p prime, n integer, denoted $\text{GF}(p^n)$.
 - Examples: $\text{GF}(3)$, $\text{GF}(8)$
- $\text{GF}(p^n)$ isomorphic to polynomials of degree $n-1$ over $\text{GF}(p)$ where addition is component-wise polynomial addition and multiplication is modulo an irreducible (no factors over $\text{GF}(p)$) polynomial over $\text{GF}(p)$ of degree n .



Example of Finite Field

- $GF(2^2)$ isomorphic to $\{p(x) = a_0 + a_1x\}$ where a_i in $GF(2) = \{0, 1\}/\text{mod } 2$.
- Addition component-wise mod 2.
 - $(x) + (1+x) = (1 + 2x) = (1)$
- Multiplication is modulo x^2+x+1 .
 - $(x) * (1+x) = (x + x^2) \text{ mod } x^2+x+1$
 - Replace x^2 by $-(x+1) = x+1$ and add
 - $(x) * (1+x) = x+1+x = 1$
 - (x) and $(1+x)$ are multiplicative inverses



Characterization of $GF(q)$

- The multiplicative group of every Galois field is cyclic. I.e., all of the non-zero elements can be represented as powers of a generator α .
 - $GF(q) = \{0, 1, \alpha, \dots, \alpha^j, \dots, \alpha^{q-2}\}$
- Every y of $GF(q)$ is root of $x^q - x$.
 - Clearly, $y = 0$ is a root. Others are roots of $x^{q-1} - 1$
 - Since $(x-1)$ is a factor of $x^{q-1} - 1$, 1 is in $GF(q)$.
 - Other elements are roots of $1 + x + x^2 + \dots + x^{q-1}$.



$(n,k,d)_q$ Reed Solomon Codes

- To encode **message** $(a_0, a_1, \dots, a_{n-1})$, a_i in $GF(q)$, evaluate $s(x) = a_0 + a_1x + \dots + a_{n-1}x^{n-1}$ for all x in $GF(q)$
- Codeword associated with $(a_0, a_1, \dots, a_{n-1})$ is $\mathbf{s} = (r(0), r(1), r(\alpha), r(\alpha^2), \dots, r(\alpha^{q-2}))$
 - Given y in $GF(q)$, the n such that $y = \alpha^n$ is the **discrete log**. It arises in cryptography.

Minimum Distance of RS Codes



- Minimum dist. of $(n, k, d)_q$ RS code is $d = n - k + 1$
 - Consider codewords \mathbf{s} and \mathbf{t} .
 - Distance between them is non-zeroes in $\mathbf{s} - \mathbf{t} = \mathbf{u}$.
 - But $u(x) = s(x) - t(x)$ is polynomial of degree $k - 1$.
 - But $u(x)$ of degree k can have at most $k - 1$ zeros.
 - Thus, $d \geq n - k + 1$.
 - But $d \leq n - k + 1$ for all $(n, k, d)_q$ codes.



Implementing RS Codes

- If Galois field is $GF(2^m)$, $(n, k, n-k+1)_q$ RS code ($q = n = 2^m$) is a $(n \log_2 n, k \log_2 n, n-k+1)_2$ binary code.
- RS codes are used on CDs and DVDs to correct against burst errors due to dust or scratches.
- Codewords can also be interlaced to help “decorrelate” errors.



Encoding RS Codes

- RS code is defined by k coefficients.

$$\begin{aligned} m(0) &= m_0 \\ m(\alpha) &= m_0 + m_1\alpha + \dots + m_k\alpha^k \\ m(\alpha^2) &= m_0 + m_1\alpha^2 + \dots + m_k\alpha^{2(k-1)} \\ &\vdots \\ m(\alpha^{q-1}) &= m_0 + m_1\alpha^{q-1} + \dots + m_k\alpha^{(q-1)(k-1)} \end{aligned}$$

- The code is linear (matrix non-sing.)

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & \alpha & \alpha^2 & \dots & \alpha^{(q-1)} \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2(q-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{q-1} & \alpha^{2(q-1)} & \dots & \alpha^{(q-1)(q-1)} \end{bmatrix} \cdot \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ \vdots \\ m_{q-1} \end{bmatrix} = \begin{bmatrix} m(0) \\ m(\alpha) \\ m(\alpha^2) \\ \vdots \\ m(\alpha^{q-1}) \end{bmatrix}$$

Decoding $(n, k, n-k+1)_q$ Reed Solomon Codes



- Let $\{\beta_j \mid 1 \leq j \leq n\}$ be elements of $GF(q)$.
- Sent codeword $\mathbf{s} = (r(\beta_1), r(\beta_2), \dots, r(\beta_n))$.
- Received word $\mathbf{r} = (\rho_1, \rho_2, \dots, \rho_n)$
- RS code can correct up to $(n-k)/2$ errors.
 - Remaining $n - (n-k)/2 = (n+k)/2$ positions correct.
- Decoding problem:
 - Given $\{(\beta_j, \rho_j) \mid 1 \leq j \leq n\}$, find polynomial $p(x)$ over $GF(q)$ with degree at most k such $p(\beta_j) = \rho_j$ for at least $(n+k)/2$ values of j .

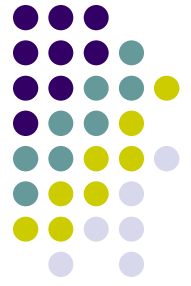


Decoding RS Codes

- Let $F = GF(p^m)$.
- The **decoding function** $D_{H,F} : F^F \rightarrow F^H \cup \{?\}$ either maps received word $\mathbf{a} = (a_1, a_2, \dots, a_{|F|})$ to a codeword $\mathbf{b} = (b_1, b_2, \dots, b_{|F|})$ at distance $\leq (|F| - |H|)/2$ from it or it maps it to “?”.

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & \alpha & \alpha^2 & \dots & \alpha^{(q-1)} \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2(q-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{q-1} & \alpha^{2(q-1)} & \dots & \alpha^{(q-1)(q-1)} \end{bmatrix}$$

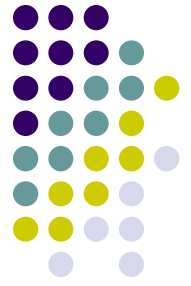
- A decoder solves system with above matrix



Extended RS Codes

- Polynomial $m(x): F \rightarrow F$ associated with $\tau: H \rightarrow F$
 - $\tau: H \rightarrow F$ is in F^H ; $m(x): F \rightarrow F$ is in F^F .
- Elements of $F = \text{GF}(p^m)$ are denoted $0, \alpha, \alpha^2, \alpha^3, \dots, \alpha^{q-1}$ where $q = p^m$.
- RS codeword associated with $\tau: H \rightarrow F$ is $(m(0), m(\alpha), \dots, m(\alpha^{q-1}))$, where $\tau(h_j) = m(h_j)$,
 - m has $|H|$ information bits, and $|F| - |H|$ check bits.
- **Encoding function** $E_{H,F} : F^H \rightarrow F^F$

Generating Extended RS Codewords



- Let $F = \text{GF}(p^m)$ and $H \subset F$ where $H = (h_1, \dots, h_{|H|})$ and $F = (f_1, \dots, f_{|F|})$
- Given $\tau: H \rightarrow F$, a function, let $m(x): F \rightarrow F$ interpolate τ over F , that is, $m(h_j) = \tau(h_j)$.

$$\begin{aligned} m(x) &= \sum_{i=1}^{|H|} \tau(h_i) \frac{\prod_{j \neq i} (x - h_j)}{\prod_{j \neq i} (h_i - h_j)} \\ &= m_0 + m_1 x + m_2 x^2 + \dots + m_{|H|-1} x^{|H|-1} \end{aligned}$$

- Note: coefficients of $m(x)$ are drawn from F .

Decoding Reed Solomon Codes



Theorem The encoding and decoding functions $E_{H,F} : F^H \rightarrow F^F$ and $D_{H,F} : F^F \rightarrow F^H \cup \{?\}$ for RS codes can be computed by circuits of size $|F| \log^{O(1)} |F|$.

Proof Due to Justesen [76] and Sarwate [77].



Error Correction Function

- It maps a received word to either “?” or to a codeword, denoted $D_{H,F}^k : F^F \mapsto F^F \cup \{?\}$
 - D 's superscript means it corrects $\leq k$ errors.

Theorem (Kaltofen-Pan) There's a randomized algorithm solving $k \times k$ Toeplitz (elements on diagonals the same) over finite field with probability $1-1/k$ in time $\log^{O(1)} k$ using $k^2 \log^{O(1)} k$ processors.

Probabilistic RS Decoding Algorithm



- It maps a received word to either “?” or to a codeword, denoted $D_{H,F}^k : F^F \mapsto F^F \cup \{?\}$
 - D 's superscript means it corrects $\leq k$ errors.

Theorem The decoding function $D_{H,F}^k$ can be computed by a randomized parallel algorithm that takes $\log^{O(1)} |F|$ time on $(k^2 + |F|) \log^{O(1)} |F|$ processors to correct $k \leq (|F| - |H|)/2$ errors. The algorithm succeeds with prob. $1 - 1/|F|$.

- Use this algorithm with $k = \sqrt{|F|}$



Generalized RS Codes

- Extend 2D RS codes to 2D **generalized RS codes** when $F = \text{GF}(2^m)$.
 - Since $F^2 = \text{GF}(2^{(m+1)})$, F^2 is also a finite field.
 $E_{H^2, F} : F^{H^2} \mapsto F^{F^2}$, $D_{H^2, F}^k : F^{F^2} \mapsto F^{F^2} \cup \{?\}$
- Encode in first dimension, then in second.
Decode in reverse order.
 - Components codeword are $a_{x,y}$ for x, y in F .
 - Can correct up to $((|F| - |H|)/2)^2$ errors, $(|F| - |H|)/2$ in each dimension separately.

Spielman's Approach to Reliable Computation



- Encode data as 2D codewords $A(x,y)$, $B(x,y)$.
- Apply polynomial $\phi(x,y)$ to each value producing a new codeword $C(x, y) = \phi(A(x,y), B(x,y))$.
- After applying ϕ , decode and re-encode each row (then column) of $C(x, y)$ separately. The result is a new codeword.
- By permuting codewords, one can simulate computation on a hypercube.