Classical linear codes

Classical linear codes: Generator matrix

- Linear code C: $[n,k]$ code
  - Encoding k bits of information into an n bit code space
  - Described by $n \times k$ generator matrix G
    - Entries $\in Z_2 = \{0, 1\}$
  - k-bit message $x \rightarrow Gx$
- Advantage of linear code
  - Compact specification
  - $[n,k]$: kn bits of general matrix
  - General encoding requires $n2^k$ bits
    - → Exponential saving

2

Examples

- $[n,k]$ code
  - Generator matrix G ($n \times k$)
  - k-bit message $x \rightarrow Gx$
- $[3,1]$ code
  - $0 \rightarrow 000$, $1 \rightarrow 111$

$$G = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \Rightarrow G[0] = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},\ G[1] = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

- $[6,2]$ code
  - $00 \rightarrow 000000$, $01 \rightarrow 000111$, …

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

3

Parity check matrix H

- $[n, k]$ code: Parity check matrix H
  - $(n\text{-}k) \times n$ matrix
  - All elements are 0, or 1 (Z_2)
- $G \rightarrow H$
  - Find $n\text{-}k$ linearly independent vectors y_i orthogonal to the columns of G
  - Set the rows of H be $y_1^T, y_2^T, \cdots y_{n\text{-}k}^T$
- $H \rightarrow G$
  - Find k linearly independent vectors z_i orthogonal to the rows of H
  - Set the columns of G be $z_1, z_2, \cdots z_k$
- $HG = 0$

$$n\text{-}k \left\{ \overbrace{\left(\quad H \quad \right)}^{n} \right. \qquad n \left\{ \overbrace{\left(\quad G \quad \right)}^{k} \right.$$

4

Parity check matrix for [3,1] code

- $[3,1]$ code

$$G = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \Rightarrow G[0] = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},\ G[1] = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad n \left\{ \overbrace{\left(\quad G \quad \right)}^{k} \right.$$

  - To construct H, pick 3-1=2 linearly independent vectors orthogonal to G

$$H \equiv \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \qquad n\text{-}k \left\{ \overbrace{\left(\quad H \quad \right)}^{n} \right.$$

  - $HG = 0$
  - Parity check

$$Hy = 0 \text{ only for } y = G[0] = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, G[1] = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \Rightarrow \boxed{\text{Error detection}}$$

5

Error detection by parity check

- How error correction works
  - Encode message x as $y = Gx$
  - Error → $y' = y \oplus e$
  - Error syndrome: $Hy' = H(y+e) = He$ ($Hy = 0$)
    - No error: $Hy' = 0$
    - Error in qubit j: He_j
      - e_j: unit vector with 1 in the j-th component
- $[3,1]$ code

$$H \equiv \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

  - Error syndrome for bit flip on bit 1, 2 & 3

$$H\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = H\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ H\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = H\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},\ H\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = H\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

6

Syndrome measurements ≡ Parity check

- [3,1] code (3-qubit bit flip code)
  - "Syndrome" measurements
    1. $Z_1 Z_2 (= Z \otimes Z \otimes I) = \begin{cases} +1 & \text{if qubit 1} = \text{qubit 2} \\ -1 & \text{if qubit 1} \neq \text{qubit 2} \end{cases}$ (Bit flip on one of the bits)
    2. $Z_2 Z_3 (= Z \otimes Z \otimes I) = \begin{cases} +1 & \text{if qubit 2} = \text{qubit 3} \\ -1 & \text{if qubit 2} \neq \text{qubit 3} \end{cases}$ (Bit flip on one of the bits)
  - Parity check

$$H \equiv \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$H\begin{bmatrix}1\\0\\0\end{bmatrix} = H\begin{bmatrix}0\\1\\1\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix},\ H\begin{bmatrix}0\\1\\0\end{bmatrix} = H\begin{bmatrix}1\\0\\1\end{bmatrix} = \begin{bmatrix}1\\1\end{bmatrix},\ H\begin{bmatrix}0\\0\\1\end{bmatrix} = H\begin{bmatrix}1\\1\\0\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix}$$

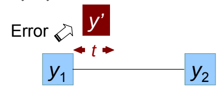
7

Global properties of the code

- Hamming distance $d(x,y)$
  - Distance between words x and b
  - number of places at which x and y differ
    - $d((1,1,0,0),(0,1,0,1)) = 2$
- Hamming weight $wt(x) = d(x,0)$
  - $wt(x+y) = d(x,y)$
- Connection to error correction
  - $y = Gx \rightarrow y' = y + e$
  - Code: Replace y' by y such that $wt(e) = d(y,y')$ is minimized
- Distance of a code C
  - Minimum distance between any two codewords

$$d(C) \equiv \min_{y_1, y_2 \in C,\, y_1 \neq y_2} d(y_1, y_2) = \min_{y \in C,\, y \neq 0} wt(y)$$

  - C: $[n, k, d]$ code
    - $d \equiv d(C) = 2t + 1$
  - → Correct errors on up to t bit



8

When error correction fails

- If $He_1 = He_2$ for $e_1 \neq e_2$ (weight $\leq t$)
  - Same syndrome for different errors
  - When e_1 occurs: $v \rightarrow v+e_1$
    - Faulty error recovery: apply e_2
      $v \rightarrow v+e_1+e_2 \neq v$
    - Message after error recovery
      $H(v+e_1+e_2) = 0$
      $\Rightarrow e_1+e_2 \in C$ (code subspace)
    - Weight of $e_1+e_2 \leq 2t$
    - If distance of the code C, $d(C) = 2t+1$, then e_1+e_2 the code cannot be in C
    - → $He_1 \neq He_2$
- → Code C with distance $d = 2t+1$ can correct errors with weight $\leq t$

9

Examples

- $[n, k, d]$ code C: $d \equiv d(C) = 2t+1$ → Correct errors on up to t bit
  - Distance d = Minimum distance between any two codewords

$$d(C) \equiv \min_{y_1, y_2 \in C,\, y_1 \neq y_2} d(y_1, y_2) = \min_{y \in C,\, y \neq 0} wt(y)$$

- [3,1] code

$$G = \begin{bmatrix}1\\1\\1\end{bmatrix} \Rightarrow G[0] = \begin{bmatrix}0\\0\\0\end{bmatrix},\ G[1] = \begin{bmatrix}1\\1\\1\end{bmatrix}$$

  - Code subspace

$$C = \left\{ \begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}1\\1\\1\end{bmatrix} \right\}$$

- [6,2] code
  - Code subspace

$$C = \left\{ \begin{bmatrix}0\\0\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\\1\\1\end{bmatrix}, \begin{bmatrix}1\\1\\1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}1\\1\\1\\1\\1\\1\end{bmatrix} \right\}$$

Distance $d(C) = 3$ → can correct 1 error

10

Hamming code

- Parity check matrix $[n, k]$
  - $n = 2^r - 1$, $k = n - r$, $r \geq 3$
- [7,4,3] code: ($r = 3$)

$$H \equiv \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

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

  - All columns are linearly independent.
  - Error syndrome He_j
  - Distance $d(C) = 3$

12

CSS (Calderbank-Shor-Steane) code – Quantum error correcting code

Construction of CSS code

- Subcode
  - Classical linear code: C_1 $[n, k_1]$
    - $(n\text{-}k_1)\times n$ Parity check matrix H_1
  - C_2 $[n, k_2]$: subcode of C_1
    - $(n\text{-}k_2)\times n$ Parity check matrix H_2
  - $k_2 < k_1$, $C_2 \subset C_1$

$$\overbrace{\begin{array}{r} n\text{-}k_1 \left\{ \vphantom{H_1} \right. \\ k_1\text{-}k_2 \left\{ \vphantom{H_1} \right. \end{array} \begin{pmatrix} H_1 \\ \cdots \end{pmatrix}}^{n} \Leftarrow H_2$$

- Equivalent relation
  - u and v (u, $v \in C_1$) are "equivalent" iff there is a $w \in C_2$ such that $u = v + w$
- CSS code:
  - $k = k_1 - k_2$ quantum code
  - a codeword = each equivalent class

$$\left|\bar{v}\right\rangle = \frac{1}{\sqrt{2^{k_2}}} \sum_{w\in C_2} \left|w+v\right\rangle \qquad v \in C_1, \notin C_2$$

$$2^{k_2} \quad 2^{k_1-k_2}$$

14

The 7-qubit (Steane) code

- Hamming code: C_1 [7,4]
  - Codewords u: spanned by the columns of G_1

$$H_1 \equiv \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \qquad G_1 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

- Subcode: C_2 [7,3]
  - Dual code $C_1^{\perp}$
    - Generator H^T, Parity check matrix G^T
  - Codewords w: spanned by the columns of G_2 ($=H_1^T$)

$$H_2 \equiv \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} (= G_1^T) \qquad G_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} (= H_1^T)$$

- CSS code ($k_1 = 4$, $k_2 = 3$, $k = k_1\text{-}k_2 = 1$)

$$\left|\bar{v}\right\rangle = \frac{1}{\sqrt{2^{k_2}}} \sum_{w\in C_2} \left|w+v\right\rangle \qquad v \in C_1, \notin C_2$$

$$2^{k_2} \quad 2^{k_1-k_2}$$

15

Codewords

- CSS code ($k_1 = 4$, $k_2 = 3$, $k = k_1\text{-}k_2 = 1$)

$$\left|\bar{v}\right\rangle = \frac{1}{\sqrt{2^{k_2}}} \sum_{w\in C_2} \left|w+v\right\rangle \qquad v \in C_1, \notin C_2$$

$$2^3 \quad 2^1$$

$$G_1 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad G_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$w_1 \quad w_2 \quad w_3$$

$$\left|\bar{0}\right\rangle = \frac{1}{\sqrt{2^{k_2}}} \sum_{w\in C_2} \left|w+0000000\right\rangle, \left|\bar{1}\right\rangle = \frac{1}{\sqrt{2^{k_2}}} \sum_{w\in C_2} \left|w+1111111\right\rangle$$

$$\left|\bar{0}\right\rangle = \frac{1}{\sqrt{8}} \begin{bmatrix} \left|0000000\right\rangle + \underset{w_1}{\left|0001111\right\rangle} + \underset{w_2}{\left|0110011\right\rangle} + \underset{w_3}{\left|1010101\right\rangle} \\ + \underset{w_1+w_2}{\left|0111100\right\rangle} + \underset{w_2+w_3}{\left|1100110\right\rangle} + \underset{w_3+w_1}{\left|1011010\right\rangle} + \underset{w_1+w_2+w_3}{\left|1101001\right\rangle} \end{bmatrix} = \frac{1}{\sqrt{8}} \sum_{\text{even } w\in \text{Hamming}} w$$

$$\left|\bar{1}\right\rangle = \frac{1}{\sqrt{8}} \begin{bmatrix} \left|1111111\right\rangle + \left|1110000\right\rangle + \left|1001100\right\rangle + \left|0101010\right\rangle \\ + \left|1000011\right\rangle + \left|0011001\right\rangle + \left|0100101\right\rangle + \left|0010110\right\rangle \end{bmatrix} = \frac{1}{\sqrt{8}} \sum_{\text{odd } w\in \text{Hamming}} w$$

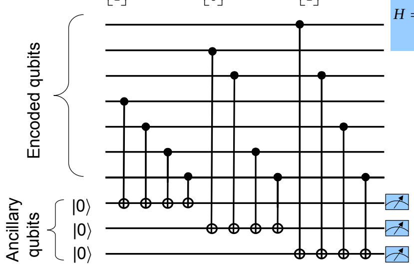
16

Bit flip error detection by parity check

- Bit flip errors
- Error syndrome = each column of H

$$H(X_1|\psi\rangle) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, H(X_2|\psi\rangle) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, H(X_3|\psi\rangle) = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \cdots$$

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

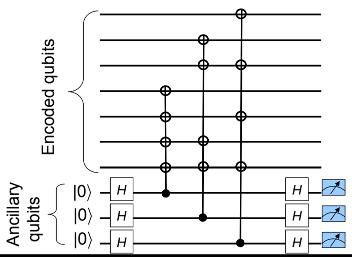


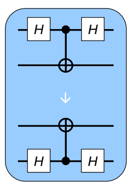
17

Phase flip error detection by parity check

- Phase flip errors
  - Change basis
  - → Bit flip errors

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





18

Logic operations on encoded qubits

Universal quantum gates

1. Single qubit gates + C-NOT (exact)
2. Hadamard + phase (S) + C-NOT + $\pi/8$ gates (T)

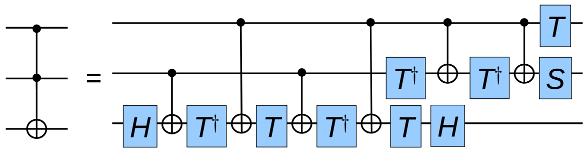
$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix},\ T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} = e^{i\pi/8} R_z\left(\frac{\pi}{4}\right)$$

- Approximate, since the set of unitary operations is continuous
- Error: $E(U,V) \equiv \max_{|\psi\rangle} \left\| (U - V)|\psi\rangle \right\| < \varepsilon$
  - U: Target unitary operator
  - V: Unitary operator implemented

20

Construction of Toffoli gate

- Using Hadamard, Phase, C-NOT, $\pi/8$ gates (Universal gate set)



21