Lecture 19

Calderbank-Shor-Steane codes

(commonly known as CSS codes)

Classical linear codes

Definition: a code is ***linear*** is the set of codewords is a linear subspace of $\{0,1\}^m$ (with respect to mod 2 arithmetic)

- XOR
- Scalar multiplication

} closed under

Examples

data	codeword
0	000
1	111

data	codeword
000	0000000
001	1010101
010	0110011
011	1100110
100	0001111
101	1011010
110	0111100
111	1101001

basis for the space

Minimum distance of a code

The **Hamming distance** between $a, b \in \{0,1\}^m$ is the number of bit positions where they differ (denoted $\Delta(a,b)$)

Distance of a code

$$d = \min_{\substack{a,b \in C \\ a \neq b}} \Delta(a,b) = \underbrace{\min_{\substack{a \in C \\ a \neq 0^m}} \Delta(a, 0^m)}_{\text{for linear codes}}$$

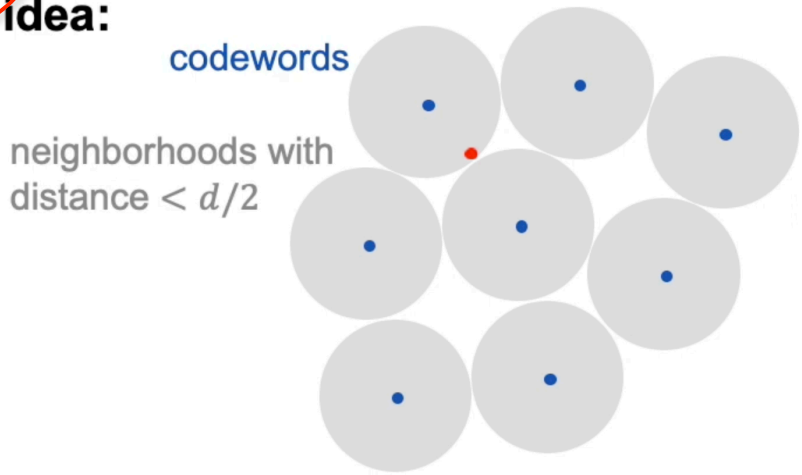
What's the minimum distance of this code?

data	codeword
000	0000000
001	1010101
010	0110011
011	1100110
100	0001111
101	1011010
110	0111100
111	1101001

Answer: $d = 4$

Theorem: if a code has distance d then up to $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors can be corrected

Proof idea:



If $< d/2$ bits of a codeword are flipped then there's a unique codeword with distance $< d/2$

Example

1001010

Dual code

Recall the **dot product** of two m-bit strings is $a \cdot b = a_1 b_1 \oplus a_2 b_2 \oplus \cdots \oplus a_m b_m$

$$= a_1 b_1 + a_2 b_2 + \cdots + a_m b_m \bmod 2$$

Definition: for any linear code $C \subseteq \{0,1\}^m$, define its ***dual code*** to be $C^\perp = \{ b \in \{0,1\}^m \mid \text{such that } b \cdot c = 0 \text{ for all } c \in C \}$

Example

$C = \{0000000,\ 1010101,\ 0110011,\ 1100110,$
$0001111,\ 1011010,\ 0111100,\ 1101001\}$

Codewords

$C^\perp = \{0000000,\ 1010101,\ 0110011,\ 1100110,$
$0001111,\ 1011010,\ 0111100,\ 1101001,$
$1111111,\ 0101010,\ 1001100,\ 0011001,$
$1110000,\ 0100101,\ 1000011,\ 0010110\}$

dual code

Generator and parity-check matrices

Generator matrix
An $n \times m$ matrix G such that, for all $a \in \{0,1\}^n$, $E(a) = aG$

Example $$[a_0 \;\; a_1 \;\; a_2] \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} = [b_0 \;\; b_1 \;\; b_2 \;\; b_3 \;\; b_4 \;\; b_5 \;\; b_6]$$

rows are a basis for C

Parity-check matrix
An $m \times (m-n)$ matrix H such that, for all $b \in \{0,1\}^n$, $b \in C$, if and only if $bH = 0$

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

columns are a basis for $C^{\perp}$

Fact: $GH = 0$

Error-correcting via parity-check matrix

Codeword $b = aG$

Codeword with error applied

$b' = b + e \pmod 2$

Error vector

$e \in \{0,1\}^m$ with Hamming weight $< d/2$

Error syndrome

$$b'H = (b + e)H$$

$$= bH + eH$$

$$= eH$$ ← ***error syndrome***

Example:

e	eH
0000000	0000
1000000	1001
0100000	1010
0010000	1011
0001000	1100
0000100	1101
0000010	1110
0000001	1111

errors with weight $< d/2$ have unique syndromes

← syndrome

correcting $1001010 \mapsto 1011010$

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

$H \otimes H \otimes \cdots \otimes H = H^{\otimes m}$ revisited

Recall $H^{\otimes m}|0^m\rangle = \sum_{b \in \{0,1\}^m} |b\rangle$ and, for all $w \in \{0,1\}^m$, $H^{\otimes m}|w\rangle = \sum_{b \in \{0,1\}^m} (-1)^{b \cdot w}|b\rangle$

Interesting generalization of these:

For any linear $C \subseteq \{0,1\}^m$, $H^{\otimes m}\left(\sum_{a \in C} |a\rangle\right) = \sum_{b \in C^\perp} |b\rangle$

and, for all $w \in \{0,1\}^m$, $H^{\otimes m}\left(\sum_{a \in C} |a + w\rangle\right) = \sum_{b \in C^\perp} (-1)^{b \cdot w}|b\rangle$

Exercise: prove these

Hint: $\sum_{a \in C} |a\rangle = \sum_{v \in \{0,1\}^n} |vG\rangle$

CSS codes: classical ingredients

Start with two nested classical linear codes $C_0 \subset C_1 \subseteq \{0,1\}^m$ such that $C_0^{\perp} \subseteq C_1$

Let d be the minimum distance of C_1

Example $C_0 = \{0000000,\ 1010101,\ 0110011,\ 1100110,\ 0001111,\ 1011010,\ 0111100,\ 1101001\}$

$C_1 = \{0000000,\ 1010101,\ 0110011,\ 1100110,\ 0001111,\ 1011010,\ 0111100,\ 1101001,\ 1111111,\ 0101010,\ 1001100,\ 0011001,\ 1110000,\ 0100101,\ 1000011,\ 0010110\}$

Generator matrices: $G_0 \in \{0,1\}^{n\times m}$ and $G_1 \in \{0,1\}^{(n+k)\times m}$

We can express $G_1 = \begin{bmatrix} G_0 \\ \hline W \end{bmatrix}$ ← $W \in \{0,1\}^{k\times m}$ the additional rows to extend the span from C_0 to C_1

Example

$$G_0 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad G_1 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad W = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

CSS codes: encoding

Start with classical linear codes with generators $G_0 \in \{0,1\}^{n\times m}$, $G_1 \in \{0,1\}^{(n+k)\times m}$, and $W \in \{0,1\}^{k\times m}$

For all $v \in \{0,1\}^k$, define the **logical** $|v\rangle$ as $|v\rangle_L = \sum_{a\in C_0} |a + vW\rangle$

So the state $\sum_{v\in\{0,1\}^k} \alpha_v |v\rangle$ gets mapped to $\sum_{v\in\{0,1\}^k} \alpha_v |v\rangle_L = \sum_{v\in\{0,1\}^k} \alpha_v \left(\sum_{a\in C_0} |a + vW\rangle\right)$

Example (Steane code)

$$|0\rangle_L = |0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle$$

} superposition of all elements of C_0

$$|1\rangle_L = |1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle$$

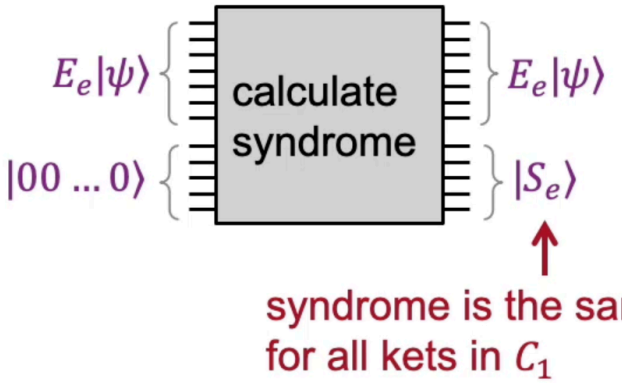
} superposition of all elements of $C_0 + 1111111$

The state $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ is encoded as $\alpha_0|0\rangle_L + \alpha_1|1\rangle_L$

CSS codes: X-error correction

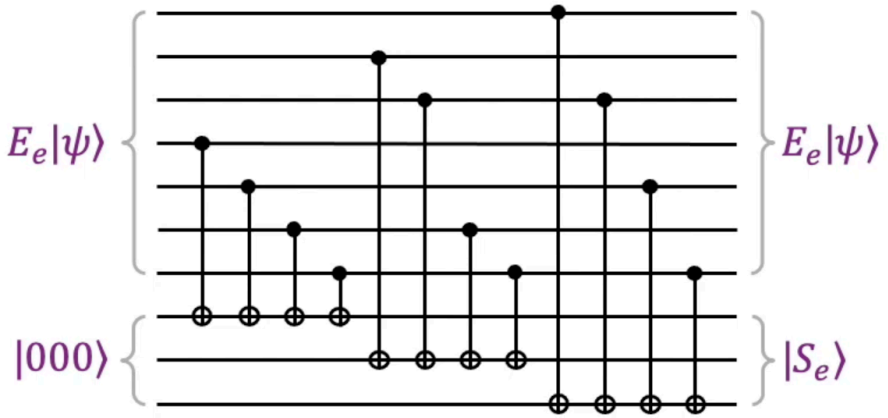
The encoding is $|\psi\rangle = \sum_{v\in\{0,1\}^k} \alpha_v \left(\sum_{a\in C_0} |a + vW\rangle \right)$ ← elements of C_1 (whose minimum distance is d)

Assume an error of the form $E_e = X^{e_0} \otimes X^{e_1} \otimes \cdots \otimes X^{e_{m-1}}$ is applied where the Hamming weight of $e = e_0 e_1 \ldots e_{m-1}$ is less than $d/2$ (this is less than $d/2$ bit flips)



Example

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



Once the syndrome has been computed, $e = e_0 e_1 \ldots e_{m-1}$ can be determined and E_e can be undone to restore $|\psi\rangle$

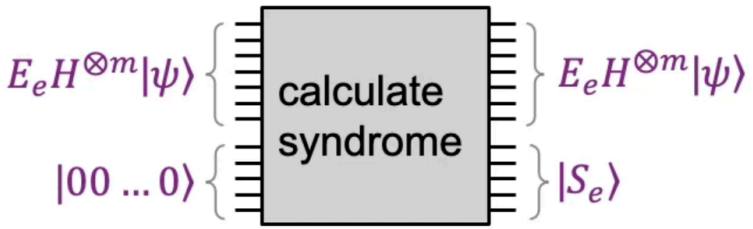
CSS codes: Z-error correction

Switch to the Hadamard basis, where Z-errors become X-errors

$$H^{\otimes m}|\psi\rangle = \sum_{v\in\{0,1\}^k} \alpha_v H^{\otimes m} \left(\sum_{a\in C_0} |a + vW\rangle \right)$$

$$= \sum_{v\in\{0,1\}^k} \alpha_v \left(\sum_{b\in C_0^\perp} (-1)^{b\cdot(vW)} |b\rangle \right) \longleftarrow \text{elements of } C_0^\perp \subseteq C_1 \text{ (whose minimum distance is } d)$$

Since $C_0^\perp \subseteq C_1$, the Z-errors can be corrected the same way as X-errors

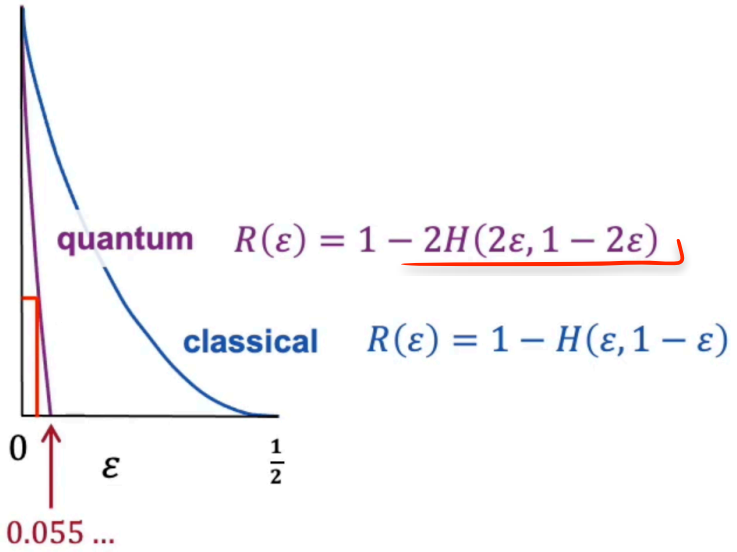


After correcting the Z-errors, apply $H^{\otimes m}$ again to switch back to the computational basis

Note that the two procedures combined also correct Y-errors because $Y = iXZ$

CSS codes: summary

CSS codes based on known classical linear codes can attain these rates as a function of ε



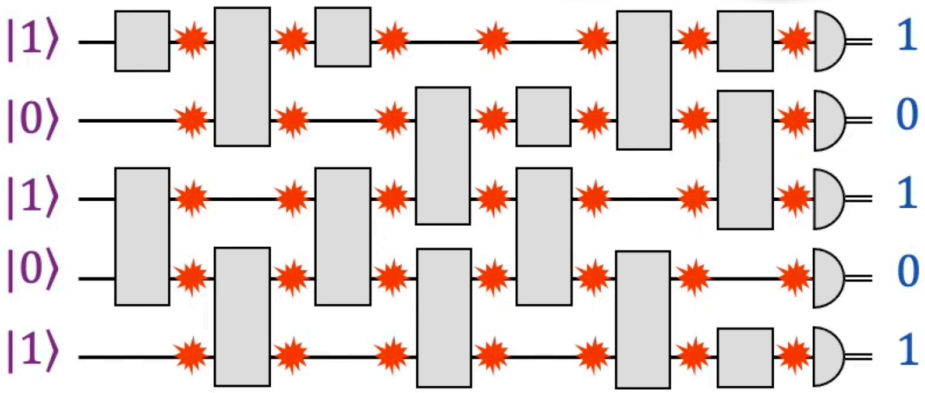
As long as $\varepsilon < 0.055$ ... there exist good quantum error-correcting codes

There are other ways of constructing quantum error-correcting codes (and other noise models)

Fault-tolerance

Quantum error-correcting codes assume errors only occur during transmission

What if there are errors ***during a computation***?



each ✸ is depolarizing channel with parameter ε

If ε is very small, this is okay—a computation of size less than $1/10\varepsilon$ succeeds most of the time

But, what if ε is constant and we want to perform arbitrarily large computations?

Threshold theorem:

There's a fixed constant $\varepsilon_0 > 0$ such that any circuit can be translated into a slightly larger** fault-tolerant circuit, that's robust against the error model with parameter $\varepsilon \leq \varepsilon_0$

** e.g., size n increases to size $O(n \log n)$