Introduction to Quantum Information Processing

QIC 710 / CS 768 / PH 767 / CO 681 / AM 871

Lecture 15 (2017)

Jon Yard
QNC 3126
jyard@uwaterloo.ca
http://math.uwaterloo.ca/~jyard/qic710

CSS Codes

Introduction to CSS codes

CSS codes (named after Calderbank, Shor, and Steane) are quantum error correcting codes that are constructed from classical error-correcting codes with certain properties

A classical ***linear*** code is one whose codewords form a subspace $C \subset \mathbb{F}_2^n$ of a vector space

In other words, the code C is closed under addition (i.e. linear combinations, as the underlying field is $\mathbb{F}_2 = \{0,1\}$ so the arithmetic is $\mathrm{mod}\ 2$).

Examples of linear codes

For $n = 7$, consider these codes (which are linear):

basis for space

$C_2 =$ {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}

Note that the minimum Hamming distance between any pair of codewords is: 4 for C_2 and 3 for C_1.

The minimum distances imply each code can correct one error

Encoding

Since , $|C_2| = 8$, it can encode 3 bits

To encode a 3-bit string $b = b_1 b_2 b_3$ in C_2, multiply b (on the right) by an appropriate 3×7 ***generator matrix***

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

Similarly, C_1 can encode 4 bits and an appropriate generator matrix for C_1 is

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

Orthogonal complement

The ***orthogonal complement of*** a linear code $C \subset \mathbb{F}_2^n$ is

$$C^\perp = \{w \in \mathbb{F}_2^n\colon\ w \cdot v = 0\ \ \forall\, v \in C\}.$$

(recall the "dot product" $w \cdot v = w_1 v_1 + \cdots + w_n v_n \bmod 2$)

Note that in the previous example, $C_2^\perp = C_1$ and $C_1^\perp = C_2$.

C_2 = {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}

We will use some of these properties in the CSS construction.

Parity check matrix

Linear codes with maximum distance d can correct up to $\left\lfloor\frac{d-1}{2}\right\rfloor$ bit-flip errors.

Every k-dimensional length-n linear code has a ***parity-check matrix*** M $(n$ by $n-k)$ such that:

- Every codeword v satisfies $vM = 0$.
- Any ***error vector*** $e \in \mathbb{F}_2^n$ with weight $\leq \left\lfloor\frac{d-1}{2}\right\rfloor$ can be uniquely determined by multiplying the disturbed codeword $v + e$ by M.

Specifically, the error e can be uniquely recovered from the **error syndrome** $s_e = (v + e)M = eM \in \mathbb{F}_2^{n-k}$.

Exercise: Find parity check matrices for C_1 and C_2.

CSS construction

Let $C_2 \subset C_1 \subset \mathbb{F}_2^n$ be two classical linear codes such that:

- The minimum distance of C_1 is d
- $C_2^\perp \subset C_1$

Let $r = \dim(C_1) - \dim(C_2) = \log\left(\frac{|C_1|}{|C_2|}\right)$

Then the resulting CSS code maps each r-qubit basis state $|b_1 b_2 \cdots b_r\rangle$ to some "coset state" of the form

$$\frac{1}{\sqrt{|C_2|}} \sum_{v \in C_2} |v + w\rangle .$$

where $w = bG \in \mathbb{F}_2^n$ is a linear function of $b \in \mathbb{F}_2^r$ chosen so that each value of w occurs in a unique coset in C_1/C_2.

The quantum code can correct up to $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors.

Example of CSS construction

For $n = 7$, for the C_1 and C_2 in the previous example we obtain these basis codewords:

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

$$|1_L\rangle = |1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle$$

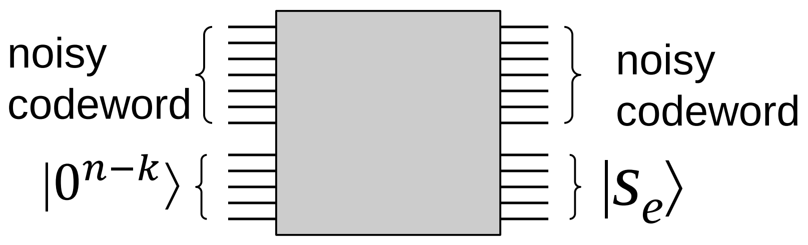
and the linear function mapping b to w can be given as $w = bG$

$$[w_1\ w_2\ w_3\ w_4\ w_5\ w_6\ w_7] = [b]\underbrace{[1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1]}_{G}$$

There is a quantum circuit that transforms between $(\alpha|0\rangle + \beta|1\rangle)|000000\rangle$ and $\alpha|0_L\rangle + \beta|1_L\rangle$

CSS error correction I

Using the error-correcting properties of C_1, one can construct a quantum circuit that computes the syndrome s for any combination of up to $\left\lfloor \frac{d-1}{2} \right\rfloor$ X-errors in the following sense



Once the syndrome s_e, has been computed, the X-errors can be determined and undone

What about Z-errors?

The above procedure for correcting X-errors has no effect on any Z-errors that occur.

CSS error correction II

Note that any Z-error is an X-error in the Hadamard basis.

Changing to Hadamard basis is like changing from C_2 to $C_2^\perp$:

$$H^{\otimes n}\left(\sum_{v\in C_2}|v\rangle\right)=\sum_{u\in C_2^\perp}|u\rangle \quad \text{and} \quad H^{\otimes n}\left(\sum_{v\in C_2}|v+w\rangle\right)=\sum_{u\in C_2^\perp}(-1)^{w\cdot u}|u\rangle\,.$$

Applying $H^{\otimes n}$ to a superposition of basis codewords yields

$$H^{\otimes n}\left(\sum_{b\in\mathbb{F}_2^r}\alpha_b\sum_{v\in C_2}|v+bG\rangle\right)=\sum_{b\in\mathbb{F}_2^r}\alpha_b\sum_{u\in C_2^\perp}(-1)^{bGu^T}|u\rangle=\sum_{u\in C_2^\perp}\sum_{b\in\mathbb{F}_2^r}\alpha_b\,(-1)^{bGu^T}|u\rangle\,.$$

Note that, since $C_2{}^\perp \subseteq C_1$, this is a superposition of elements of C_1, so we can use the error-correcting properties of C_1 to correct.

Then, applying Hadamards again, restores the codeword with up to d Z-errors corrected

CSS error correction III

The two procedures together correct up to $\left\lfloor\frac{d-1}{2}\right\rfloor$ X-errors that and up to $\left\lfloor\frac{d-1}{2}\right\rfloor$ Z-errors. Since $Y = iXZ$, this means they can correct $\left\lfloor\frac{d-1}{2}\right\rfloor$ Y-errors.

From this, a simple linearity argument can be applied to show that the code corrects up to $\left\lfloor\frac{d-1}{2}\right\rfloor$ arbitrary errors (that is, the error can be any quantum operation performed on up to $\left\lfloor\frac{d-1}{2}\right\rfloor$ qubits).

Since there exist pretty good *classical* codes that satisfy the properties needed for the CSS construction, this approach can be used to construct pretty good *quantum* codes.

Depolarizing channel

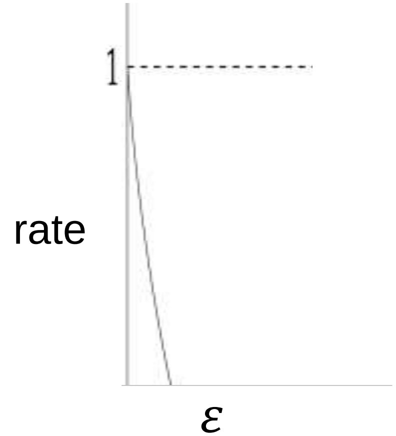
Each qubit incurs the following type of error ($0 \leq \varepsilon \leq 1$):

$$\begin{cases} I & \text{with probability } 1 - 3\varepsilon/4 & \text{(no error)} \\ X & \text{with probability } \varepsilon/4 & \text{(bit flip)} \\ Z & \text{with probability } \varepsilon/4 & \text{(phase flip)} \\ Y & \text{with probability } \varepsilon/4 & \text{(both)} \end{cases}$$

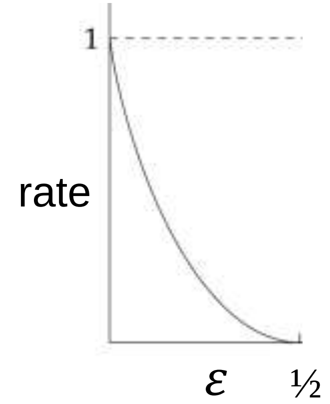
For any noise rate below $\varepsilon \approx .255$ (whether this can go as high as $\varepsilon = 1/3$ is a major open question), there are codes with:

- finite rate (message expansion by a constant factor: $R = k/n$)
- error probability approaching zero as $n \to \infty$.

depolarizing channel

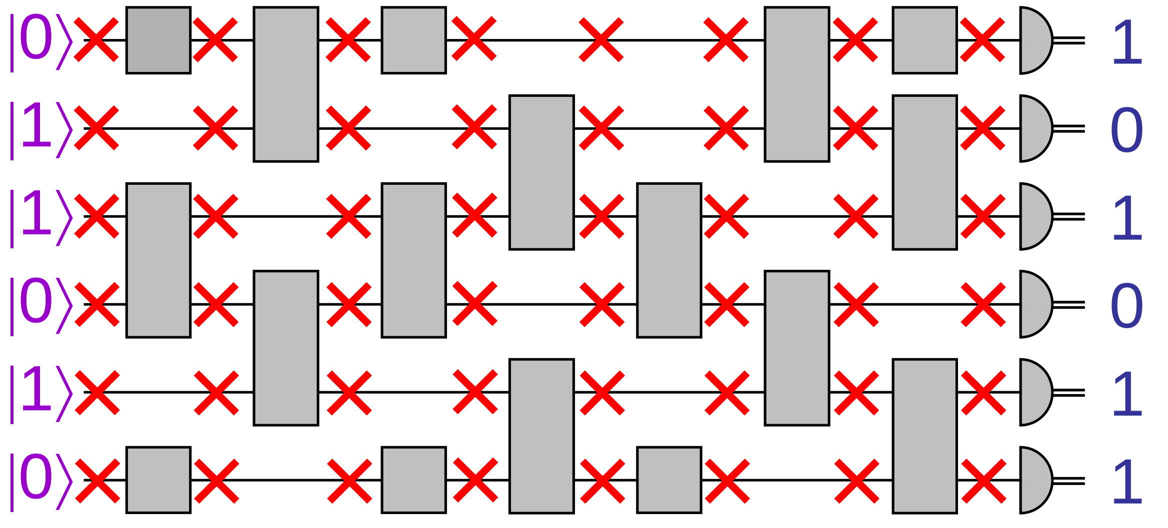


binary symmetric channel



Brief remarks about fault-tolerant computing

A simple error model



At each qubit there is an ✕ error per unit of time, that denotes the following noise:

$$\begin{cases} I & \text{with probability } 1 - 3\varepsilon/4 \\ X & \text{with probability } \varepsilon/4 \\ Z & \text{with probability } \varepsilon/4 \\ Y & \text{with probability } \varepsilon/4 \end{cases}$$

Threshold theorem

If ε is very small then this is okay — a computation of size* less than $O\left(\frac{1}{\varepsilon}\right)$ will still succeed most of the time.

But, for every ***constant*** value of ε, the size of the maximum computation possible in this manner is constant

Threshold theorem:

There's a ***fixed*** constant $\varepsilon_0 > 0$ such that a circuit of ***any*** size T can be translated into a circuit of size $O(T\log^c(T))$ that is robust against the error model with parameter $\varepsilon \leq \varepsilon_0$.

(The proof is omitted here)

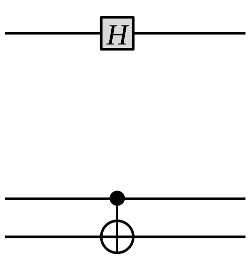
* where size = (# qubits)x(# time steps)

Comments about the threshold theorem

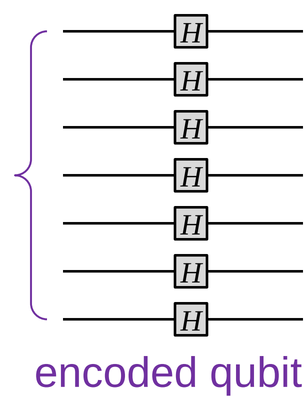
Idea is to use a quantum error-correcting code at the start and then perform all the gates ***on the encoded data***

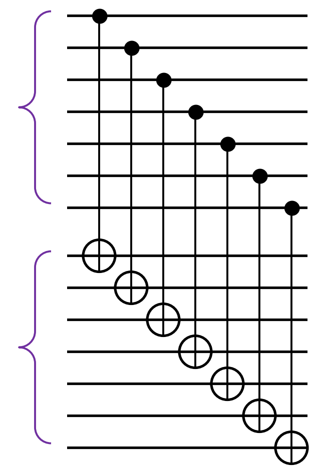
At regular intervals, an error-correction procedure is performed, very carefully, since these operations are also subject to errors! (Need to correct errors faster than they are created)

The 7-qubit CSS code has some nice properties that enable gates from the Clifford group (e.g. H and CNOT) to be directly performed on the encoded data "transversally" in the sense that:



are equivalent to





Also, codes applied recursively become stronger