Introduction to Quantum Information Processing
QIC 710 / CS 768 / PH 767 / CO 681 / AM 871

Lecture 15 (2016)

Richard Cleve
QNC 3129
cleve@cs.uwaterloo.ca

© Richard Cleve 2016
Classical error correcting codes
Binary symmetric channel

Each bit that goes through it has probability $\varepsilon$ of being flipped.

**3-bit repetition code:**

- Encode each bit $b$ as $bbb$
- Decode each received message $b_1b_2b_3$ as majority($b_1, b_2, b_3$)

This reduces the effective error probability per data bit to $3\varepsilon^2$ at a cost of tripling the message length (“rate” is $1/3$).

**Is this useful?**

- If $\varepsilon = 0.1$ and this is applied to $n$-bit messages then around $3\%$ of the $n$ bits will be in error, rather than $10\%$
- If $\varepsilon = 0.01$ and this is applied to $n$-bit messages then around $0.03\%$ of the $n$ bits will be in error, rather than $1\%$

**Can one do better?**
A rough “big picture” view I

“Good” codes (for classical information):

Message: 0100110101110101 (some $n$-bit string)

Encoding: 0110011010100101111101010111010 (m bits) constant expansion

Errors: 0100111010110110110110110110110

Decoding: 0100110101110101 no errors with probability $\to 1$ as $n \to \infty$

$n/m$ is the rate of the code ( = reciprocal of message expansion)

Theorem (good codes exist):
For all $\varepsilon < 1/2$, there exist encoding and decoding functions $E:\{0,1\}^n \to \{0,1\}^m$ and $D:\{0,1\}^m \to \{0,1\}^n$ such that $n/m$ is constant and the probability of any errors $\to 0$ as $n \to \infty$
A rough “big picture” view II

Rate as a function of noise level:

Each bit going through channel flips with probability $\varepsilon$

$$R(\varepsilon) = 1 - H(\varepsilon, 1 - \varepsilon) = 1 - (-\varepsilon \log(\varepsilon) - (1 - \varepsilon) \log(1 - \varepsilon))$$

The optimal rate $R$ (reciprocal of message expansion) for which there exist “good” codes as a function of $\varepsilon$

Informally, “good” codes with block length $n$ have this property:

probability of any errors occurring in block $\to 0$ as $n \to \infty$

What about quantum error correcting codes?
Shor's 9-qubit code
3-qubit code for one $X$-error

The following 3-qubit quantum code protects against up to one error, *if* the error can only be a quantum bit-flip (an $X$ operation)

\[
\alpha|0\rangle + \beta|1\rangle \quad \text{encode} \quad e \quad \text{error} \quad \alpha|0\rangle + \beta|1\rangle \quad \text{decode}
\]

Error can be any one of: $I \otimes I \otimes I \quad X \otimes I \otimes I \quad I \otimes X \otimes I \quad I \otimes I \otimes X$

Corresponding syndrome: $|00\rangle \quad |11\rangle \quad |10\rangle \quad |01\rangle$

The essential property is that, in each case, the data $\alpha|0\rangle + \beta|1\rangle$ is shielded from (i.e., unaffected by) the error.

What about $Z$ errors? This code leaves them intact …
3-qubit code for one Z-error

Using the fact that $HZH = X$, one can adapt the previous code to protect against $Z$-errors instead of $X$-errors.

$$\alpha|0\rangle + \beta|1\rangle \quad H \quad H \quad e \quad H \quad H \quad \alpha|0\rangle + \beta|1\rangle$$

$|0\rangle$ encode $\text{error} |S_e\rangle$

$|0\rangle$ decode

Error can be any one of: $I \otimes I \otimes I$, $Z \otimes I \otimes I$, $I \otimes Z \otimes I$, $I \otimes I \otimes Z$

This code leaves $X$-errors intact

Is there a code that protects against errors that are arbitrary one-qubit unitaries?
Shor’s 9-qubit quantum code

The “inner” part corrects any single-qubit $X$-error
The “outer” part corrects any single-qubit $Z$-error
Since $Y = iZX$, single-qubit $Y$-errors are also corrected
Arbitrary one-qubit errors

Suppose that the error is some arbitrary one-qubit unitary operation $U$

Since there exist scalars $\lambda_1, \lambda_2, \lambda_3$ and $\lambda_4$, such that

$$U = \lambda_1 I + \lambda_2 X + \lambda_3 Y + \lambda_4 Z$$

a straightforward calculation shows that, when a $U$-error occurs on the $k^{th}$ qubit, the output of the decoding circuit is

$$(\alpha |0\rangle + \beta |1\rangle)(\lambda_1 |s_{e_1}\rangle + \lambda_2 |s_{e_2}\rangle + \lambda_3 |s_{e_3}\rangle + \lambda_4 |s_{e_4}\rangle)$$

where $s_{e_1}, s_{e_2}, s_{e_3}$ and $s_{e_4}$ are the syndromes associated with the four errors ($I, X, Y$ and $Z$) on the $k^{th}$ qubit

Hence the code actually protects against any unitary one-qubit error (in fact the error can be any one-qubit quantum operation)
Summary of 9-qubit code

Can recover data from \textit{any} 1 qubit error:

It turns out the data can also be recovered data from \textit{any} 2 qubit \textit{erasure} error:
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 (a subset of \( \{0,1\}^m \)) constitute a vector space.

In other words, they are closed under linear combinations (here the underlying field is \( \{0,1\} \) so the arithmetic is \( \text{mod } 2 \)).
Examples of linear codes

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

$C_2 = \{0000000, \ 1010101, \ 0110011, \ 1100110, \ 0001111, \ 1011010, \ 0111100, \ 1101001\}$

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

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 \(b = b_1b_2b_3\) in \(C_2\), one multiplies \(b\) (on the right) by an appropriate 3×7 generating matrix

\[
G = \begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1
\end{bmatrix}
\]

Similarly, \(C_1\) can encode 4 bits and an appropriate generating matrix for \(C_1\) is

\[
\begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
\]
Orthogonal complement

For a linear code $C$, define its **orthogonal complement** as

$$C^\perp = \{ w \in \{0,1\}^m : \text{for all } v \in C, w \cdot v = 0 \}$$

(where $w \cdot v = \sum_{j=1}^{m} w_j v_j \mod 2$, the “dot product”)

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

- For every codeword $v$, $vM = 0$
- For any **error-vector** $e \in \{0,1\}^m$ with weight $\leq \left\lfloor \frac{d-1}{2} \right\rfloor$, $e$ can be uniquely determined by multiplying the disturbed codeword (which is $v+e$) by $M$

**Error syndrome:** $(v+e)M = s_e$ and $e$ is a function of $s_e$ only

**Exercise:** determine the parity check matrix for $C_1$ and for $C_2$
CSS construction

Let $C_2 \subset C_1 \subset \{0,1\}^m$ be two classical linear codes such that:

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

Let $r = \dim(C_1) - \dim(C_2) = \log(|C_1|/|C_2|)$

Then the resulting CSS code maps each $r$-qubit basis state $|b_1...b_r\rangle$ to some “coset state” of the form

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

where $w = w_1...w_m$ is a linear function of $b_1...b_r$ chosen so that each value of $w$ occurs in a unique coset in the quotient space $C_1/C_2$

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

For \( m = 7 \), for the \( C_1 \) and \( C_2 \) in the previous example we obtain these basis codewords:

\[
|0_L\rangle = |0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle \\
\quad \quad + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle
\]

\[
|1_L\rangle = |1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle \\
\quad \quad + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle
\]

and the linear function mapping \( b \) to \( w \) can be given as \( w = b \cdot G \)

\[
\begin{bmatrix}
w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7
\end{bmatrix} = \begin{bmatrix} b \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}
\]

There is a quantum circuit that transforms between \( (\alpha |0\rangle + \beta |1\rangle) |0^{m-1}\rangle \) and \( \alpha |0_L\rangle + \beta |1_L\rangle \)
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 $d$ $X$-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 $C_1$ since

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

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

$$H^\otimes m\left(\sum_{b \in \{0,1\}^r} \sum_{v \in C_2} |v + b \cdot G\rangle\right) = \sum_{b \in \{0,1\}^r} \sum_{u \in C_2^\perp} (-1)^{b \cdot G \cdot u} |u\rangle = \sum_{u \in C_2^\perp} \sum_{b \in \{0,1\}^r} \alpha_b (-1)^{b \cdot G \cdot u} |u\rangle$$

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 $d$ errors that can each be either an $X$-error or a $Z$-error — and, since $Y = iXZ$, they can also be $Y$-errors.

From this, a simple linearity argument can be applied to show that the code corrects up to $d$ arbitrary errors (that is, the error can be any quantum operation performed on up to $d$ 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 ($0 \leq \varepsilon \leq \frac{3}{4}$):

\[
\begin{align*}
    I & \quad \text{with probability } 1 - \varepsilon \quad \text{(no error)} \\
    X & \quad \text{with probability } \frac{\varepsilon}{3} \quad \text{(bit flip)} \\
    Z & \quad \text{with probability } \frac{\varepsilon}{3} \quad \text{(phase flip)} \\
    Y & \quad \text{with probability } \frac{\varepsilon}{3} \quad \text{(both)}
\end{align*}
\]

For any noise rate below some constant, there are codes with:

- finite rate (message expansion by a constant factor: $r = \frac{n}{m}$)
- error probability approaching zero as $n \to \infty$
Brief remarks about fault-tolerant computing
A simple error model

At each qubit there is an \( \times \) error per unit of time, that denotes the following noise:

\[
\begin{align*}
|0\rangle & \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad 1 \\
|1\rangle & \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad 0 \\
|1\rangle & \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad 1 \\
|0\rangle & \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad 0 \\
|1\rangle & \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad 1 \\
|0\rangle & \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad 1
\end{align*}
\]

\[
\begin{align*}
I & \quad \text{with probability} \quad 1 - \varepsilon \\
X & \quad \text{with probability} \quad \varepsilon / 3 \\
Y & \quad \text{with probability} \quad \varepsilon / 3 \\
Z & \quad \text{with probability} \quad \varepsilon / 3
\end{align*}
\]
Threshold theorem

If $\varepsilon$ is very small then this is okay—a computation of size* less than $1/(10\varepsilon)$ will still succeed most of the time

But, for every constant value of $\varepsilon$, 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!

The 7-qubit CSS code has some nice properties that enable some (not all) gates to be directly performed on the encoded data: $H$ and $CNOT$ gates act “transversally” in the sense that:

- $H$ are equivalent to

- $H$ are equivalent to

Also, codes applied recursively become stronger.