Introduction to quantum information processing

CSS codes

Brad Lackey

10 November 2016

OUTLINE

Last time...

- The three bit repetition code encodes $\alpha|0\rangle + \beta|1\rangle \mapsto \alpha|000\rangle + \beta|111\rangle$.
- It's check operators are $S_1 = Z_1 Z_2$ and $S_2 = Z_2 Z_3$, with correction in the table below.
- The phase flip channel has Kraus form $\mathcal{P}(\rho) = (1-p)\rho + pZ\rho Z$.
- The fidelity has formula $F(\rho,\sigma) = \sqrt{\sigma^{\frac{1}{2}}\rho\sigma^{\frac{1}{2}}}$.
- If $\sigma = |\psi\rangle\langle\psi|$ is pure this is easier to evaluate:

$$F(\rho, |\psi\rangle\langle\psi|) = \sqrt{\langle\psi|\rho|\psi\rangle}.$$

Syndrome measurment	Correction operator
$(+1,+1)$	$\mathbb{1}$
$(-1,+1)$	X_1
$(+1,-1)$	X_3
$(-1,-1)$	X_2

Outline

Three qubit code

Phase flip errors

Suppose quantum information is encoded in the 3-qubit repetition code:

$$|\psi_{\text{enc}}\rangle = \alpha|000\rangle + \beta|111\rangle$$

Consider instead, the phase flip channel on each qubit as the error model:

$$\mathcal{P}_p^{\otimes 3}(\rho) = (1-p)^3\rho + p(1-p)^2(Z_1\rho Z_1 + Z_2\rho Z_2 + Z_3\rho Z_3) + p^2(1-p)(Z_1Z_2\rho Z_1Z_2 + Z_2Z_3\rho Z_2Z_3 + Z_1Z_3\rho Z_1Z_3) + p^3 Z_1Z_2Z_3\rho Z_1Z_2Z_3.$$

For example the error operator $\propto Z_1$ has

$$G_{\text{dec}}Z_1(\alpha|000\rangle + \beta|111\rangle) = G(\alpha|000\rangle - \beta|111\rangle) = (\alpha|0\rangle - \beta|1\rangle) \otimes |00\rangle.$$

The other errors are similar: now taking $|\phi\rangle = \alpha|0\rangle - \beta|1\rangle$

$$G_{\text{dec}}\mathcal{P}_p^{\otimes 3}(\rho)G_{\text{dec}}^\dagger = |\psi\rangle\langle\psi| \otimes \left((1-p)^3 + 3p^2(1-p)\right)|00\rangle\langle 00| + |\phi\rangle\langle\phi| \otimes \left(3p(1-p)^2 + p^3\right)|00\rangle\langle 00|.$$

Now, $|\langle\psi|\phi\rangle|^2 = |\alpha|^2 - |\beta|^2 = 0$ when $|\alpha| = |\beta|$, so

$$F(\mathfrak{C}, \mathcal{P}_p^{\otimes 3}) = \min_{|\psi\rangle} \sqrt{(1-p)(1-2p+4p^2) + O(p)(|\alpha|^2 - |\beta|^2)} = \sqrt{(1-p)(1-2p+4p^2)}.$$

A ROTATED THREE QUBIT CODE

Why does the 3-qubit code do poorly on phase errors?

- The code is designed so that bit flips move out of the code space.
- A phase error doesn't: $\alpha|000\rangle + \beta|111\rangle \mapsto \alpha|000\rangle - \beta|111\rangle$.

Here is an important circuit identity: $HZH = X$.

- This converts phase errors Z into bit errors X.
- Recall: $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = H|0\rangle$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = H|1\rangle$
- One can check directly: $Z|+\rangle = |-\rangle$ and $Z|-\rangle = |+\rangle$.
- So in the $\{|+\rangle, |-\rangle\}$ basis a phase error acts like a bit flip!

We can protect against phase flip error by "rotating" the code with H:

$$\tilde{\mathfrak{C}} = \{|+++\rangle, |---\rangle\}.$$

The phase error check operators are also rotated: $\tilde{S}_1 = X_1X_2$ and $\tilde{S}_2 = X_2X_3$.

But now bit flips act like phase flip errors on $\tilde{\mathfrak{C}}$. Argh!

The Shor code

Shor (1995) came up with a way to protect against both bit and phase flips.

- First protect against phase error using the rotated three qubit code:

$$|0\rangle \mapsto |+++\rangle \text{ and } |1\rangle \mapsto |---\rangle.$$

- Then protect each of these three qubits against bit flips:

$$|+\rangle \mapsto \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \text{ and } |-\rangle \mapsto \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle).$$

So all told:

$$|0\rangle \mapsto \frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle)$$
$$|1\rangle \mapsto \frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle).$$

In (quantum) coding theory language, the Shor code has

- length $n = 9$ and dimension $k = 1$. I.e. $\dim \mathfrak{H} = 2^9$ and $\dim \mathfrak{C} = 2^1$.

SHOR CODE SYNDROME MEASUREMENTS

Let's look at how the Shor code works via it's syndromes.

- The 3 qubit code has check operators $S_1 = Z_1Z_2$ and $S_2 = Z_2Z_3$.
- The Shor code has three such codes. For bit-flips one checks:

$$S_1 = Z_1Z_2,\ S_2 = Z_2Z_3,\ S_3 = Z_4Z_5,\ S_4 = Z_5Z_6,\ S_5 = Z_7Z_8,\ S_6 = Z_8Z_9.$$

The rotated 3 qubit code has check operators $\tilde{S}_1 = X_1X_2$ and $\tilde{S}_2 = X_2X_3$.

- But each $|\pm\rangle$ is now three qubits $|\pm\rangle \mapsto \frac{1}{\sqrt{2}}(|000\rangle \pm |111\rangle)$. Note:

$$Z_j(|000\rangle + |111\rangle) = |000\rangle - |111\rangle \text{ for any } j = 1, 2, 3.$$

- So $X_1X_2X_3$ checks to see if *any* phase flip has happened.
- Why is this? $\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle$ is an eigenvector with syndrome -1.
- Therefore the check operators for phase errors are:

$$\tilde{S}_1 = X_1X_2X_3X_4X_5X_6 \text{ and } \tilde{S}_2 = X_4X_5X_6X_7X_8X_9.$$

Shor code error correction

Bit errors are easy to identify with the given syndrome operators:

$$\underbrace{S_1 = Z_1Z_2,\ S_2 = Z_2Z_3,}_{X\text{-error among qubits } 1,\,2,\,3}\ \underbrace{S_3 = Z_4Z_5,\ S_4 = Z_5Z_6,}_{X\text{-error among qubits } 4,\,5,\,6}\ \underbrace{S_5 = Z_7Z_8,\ S_6 = Z_8Z_9}_{X\text{-error among qubits } 7,\,8,\,9}.$$

In fact: the Shor code can correct one bit error in each of these three blocks:

- Upon measuring a syndrome: apply the appropriate X-correction.

A phase error checks a bit differently:

$$\underbrace{\tilde{S}_1 = X_1X_2X_3X_4X_5X_6,}_{Z\text{-error among qubits } 1-6}\ \underbrace{\tilde{S}_2 = X_4X_5X_6X_7X_8X_9}_{Z\text{-error among qubits } 4-9}.$$

Syndrome measurment	Correction operator
$(+1,+1)$	$\mathbb{1}$
$(-1,+1)$	Z_1
$(+1,-1)$	Z_7
$(-1,-1)$	Z_4

The checks only determine the block, but

- a single correction fixes all possibilities!

X and Z checks are independent: $Y = iXZ$ errors are corrected too.

OUTLINE

Classical codes for bit errors

The Shor code promises to correct one bit error.

- It can correct up to three: as long as they are in different blocks.

The Shor code promises to correct one phase error.

- It can correct up to three: as long as they are in the same block.

This is a lot of extra capability.

- Is there a smaller code that just corrects one bit and one phase error?

Correcting bit errors is just classical error correction.

- Let C_1 be a (linear) $[n, k_1]$ code (n = length and k_1 = dimension).
- I.e. $C_1 \subset \mathbb{F}_2^n$ is a k_1-dimensional subspace.
- Suppose the code can correct t bit flip errors: i.e.
- an algorithm exists to cancel any error $\vec{e}$ of $|\vec{e}| \leq t$ from a codeword.

Operationally:

$$\vec{w} \xrightarrow{\text{Tranmission}} \vec{w} \oplus \vec{e} \xrightarrow{\text{Correction}} \vec{w}$$

Dual codes for phase errors

It turns out correcting phase errors is related to a "dual" code.

- Basic idea: Hadamard gates switch between a code and its dual,
- *and* it switches phase errors for bit errors.

Let C_2 be a (linear) $[n, k_2]$ code (again n = length and k_2 = dimension).

- Write $C_2^\perp = \{\vec{w} \in \mathbb{F}_2^n \ : \ \vec{v} \cdot \vec{w} = 0 \pmod 2 \text{ for all } w \in C_2\}$.
- *Fact*: if $\vec{v} \notin C_2^\perp$ then $\vec{v} \cdot \vec{w} = 1$ for exactly $2^{k_2 - 1}$ vectors in C_2.
- In particular,

$$\sum_{w \in C_2} (-1)^{\vec{v}\cdot\vec{w}} = \begin{cases} 2^{k_2} & \text{if } \vec{v} \in C_2^\perp \\ 0 & \text{if } \vec{v} \notin C_2^\perp. \end{cases}$$

Specifically, using Hadamard gates

$$H^{\otimes n} \left(\frac{1}{2^{k_2/2}} \sum_{\vec{w} \in C_2} |\vec{w}\rangle \right) = \frac{1}{2^{(n+k_2)/2}} \sum_{\vec{v} \in \mathbb{F}_2^n} \sum_{\vec{w} \in C_2} (-1)^{\vec{v}\cdot\vec{w}} |\vec{v}\rangle = \frac{1}{2^{(n-k_2)/2}} \sum_{\vec{v} \in C_2^\perp} |\vec{v}\rangle.$$

THE CSS-CONSTRUCTION

The Calderbank-Shor-Steane (CSS) construction converts to classical codes into a single quantum code. The starting point is:

1. C_1 is a $[n, k_1]$-code (i.e. encodes k_1-bits with n-bits),
2. $C_2 \subseteq C_1$ is a $[n, k_2]$-code (so $k_2 \leq k_1$), and
3. C_1 and $C_2^{\perp}$ both correct t errors.

We encode $k_1 - k_2$ qubits with n-qubits as elements of C_1/C_2. I.e. a basis of our quantum code is

$$|\vec{x} + C_2\rangle = \frac{1}{2^{k_2/2}} \sum_{\vec{w} \in C_2} |\vec{x} + \vec{w}\rangle \text{ for } \vec{x} \in C_1.$$

Theorem (Calderbank-Shor)

This code will correct t bit-flip and t phase-flip errors.

Consequently it corrects arbitrary errors on t qubits. (Assignment 5, #3(b).)

THE CSS-CONSTRUCTION

Proof: Let $\vec{e}_1$ and $\vec{e}_2$ be vectors of Hamming weight at most t. Then:

$$|\vec{x} + C_2\rangle \xrightarrow{\text{error}} \frac{1}{2^{k_2/2}} \sum_{\vec{w} \in C_2} (-1)^{(\vec{x}+\vec{w})\cdot\vec{e}_2} |\vec{x} + \vec{w} + \vec{e}_1\rangle$$

$$\xrightarrow{C_1\text{-correct}} \frac{1}{2^{k_2/2}} \sum_{\vec{x} \in C_2} (-1)^{(\vec{x}+\vec{w})\cdot\vec{e}_2} |\vec{x} + \vec{w}\rangle$$

$$\xrightarrow{H^{\otimes}n} \frac{1}{2^{(n+k_2)/2}} \sum_{\vec{z} \in \mathbb{F}_2^n} \sum_{y \in C_2} (-1)^{(\vec{x}+\vec{w})\cdot(\vec{z}+\vec{e}_2)} |\vec{z}\rangle$$

$$= \frac{1}{2^{(n+k_2)/2}} \sum_{\vec{z} \in \mathbb{F}_2^n} \sum_{\vec{w} \in C_2} (-1)^{(\vec{x}+\vec{w})\cdot\vec{z}} |\vec{z} \oplus \vec{e}_2\rangle = \frac{1}{2^{(n-k_2)/2}} \sum_{\vec{z} \in C_2^\perp} (-1)^{\vec{x}\cdot\vec{z}} |\vec{z} \oplus \vec{e}_2\rangle$$

$$\xrightarrow{C_2^\perp\text{-correct}} \frac{1}{2^{(n-k_2)/2}} \sum_{\vec{z} \in C_2^\perp} (-1)^{\vec{x}\cdot\vec{z}} |\vec{z}\rangle$$

$$\xrightarrow{H^{\otimes n}} \frac{1}{2^{k_2/2}} \sum_{\vec{w} \in C_2} |\vec{x} + \vec{w}\rangle = |\vec{x} + C_2\rangle. \quad \square$$

Outline

The Steane code

For readability sake we will focus on the $[7,4,3]$ Hamming code:

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

with dual code

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

The Steane code is formed by using the CSS-construction:

$$\begin{aligned} |0_L\rangle &\mapsto \frac{1}{\sqrt{8}}(|0000000\rangle + |0001111\rangle + |0110011\rangle + |0111100\rangle \\ &\quad + |1010101\rangle + |1011010\rangle + |1100110\rangle + |1101001\rangle). \\ |1_L\rangle &\mapsto \frac{1}{\sqrt{8}}(|1111111\rangle + |1110000\rangle + |1001100\rangle + |1000011\rangle \\ &\quad + |0101010\rangle + |0100101\rangle + |0011001\rangle + |0010110\rangle). \end{aligned}$$

Next time...

- The stabilizer formalism.