Stabilizer Codes, cont.

Note Title 5/31/2012

Given a Stabilizer $S = \langle g_1, \ldots, g_e \rangle$, we want to find a basis of our code space V_S.
We would also like to determine which types of errors we can correct and how to do so.

Recall the theorem that states conditions for when a set of errors can be corrected:

Theorem: There is an error correcting code for codewords spanned by $\{|\psi_j\rangle\}$ that corrects errors spanned by $\{E_a\}$ if and only if

$$\langle \psi_i | E_a^\dagger E_b | \psi_j \rangle = \delta_{ij} C_{ab} \quad \forall a, b, i, j$$

for Hermitian matrix C.

We will derive a new version of these conditions translated into the language of stabilizers. First a definition:

Given S a subgroup of group G_n Define subgroup $N(S)$ to be all elements $E \in G_n$ such that $EgE^\dagger \in S \quad \forall g \in S$.

Note that $S \subseteq N(S) \subseteq G_n$.

Theorem (Error-correcting conditions for Stabilizer codes)
Let S be a stabilizer for stabilizer code $C(S)$.
Suppose $\{E_j\}$ is a set of operators in G_n such that.
$E_j^\dagger E_k \notin N(S) - S \quad \forall j, k$. Then $\{E_j\}$ is a correctable set of errors for $C(S)$.

Proof: Let $\{|\psi_k\rangle\}$ be a basis for V_S. There are two cases:

Case 1: $E_j^\dagger E_k \in S$. Then $\langle \psi_l | E_j^\dagger E_k | \psi_m \rangle = \delta$

($E_j^\dagger E_k |\psi_m\rangle = |\psi_m\rangle$)

Case 2: $E_j^\dagger E_k \notin N(S)$ Then $\exists\ g \in S$ s.t.
$E_j^\dagger E_k$ g $(E_j^\dagger E_k)^\dagger \notin S$.

In particular g and $E_j^\dagger E_k$ anti-commute.

$$\langle \psi_l | E_j^\dagger E_k\, g | \psi_m \rangle = - \langle \psi_l | g\, E_j^\dagger E_k | \psi_m \rangle$$

$$\langle \psi_l | E_j^\dagger E_k | \psi_m \rangle \qquad - \langle \psi_l | E_j^\dagger E_k | \psi_m \rangle$$

$$= 0$$

The conditions for the first Theorem (general conditions for ECC) are satisfied. //

How do we actually correct the error?

Measure the generators of the stabilizer to get $f_1, f_2, \ldots, f_e$
(f_i is the eigenvalue of g_i obtained from the measurement).

Use classical computation to determine the error E_j that occurred and then apply $E_j^\dagger$ to the state.

Let β_{jk} satisfy: $E_j g_k E_j^\dagger = \beta_{jk} g_k$

Store β_{jk}'s in a table

$\vec{\beta}_j = (\beta_{j1} \ldots \beta_{je})$

Find $\vec{\beta}_j$ that matches measured syndrome.

Note that because of the ECC conditions above, the measurement results are independent of the code state $|\psi\rangle$ + depend only on the error that occurred.

What if more than one error results in the same error syndrome?

Suppose $E_j g_k E_j^\dagger = E_{j'} g_k E_{j'}^\dagger$ for all k.

This means $E_{j'}^\dagger E_j g_k = g_k E_{j'}^\dagger E_j \quad \forall k$

(i.e. $E_{j'}^\dagger E_j$ commutes with all of S).

Thus $E_{j'}^\dagger E_j \in N(S)$, but by assumption, it can't be in $N(S) - S$., so $E_{j'}^\dagger E_j \in S$.

This means we can use $E_{j'}^\dagger$ to correct error E_j.

$E_{j'}^\dagger E_j |\psi\rangle = |\psi\rangle$.

So given an error syndrome $\vec{\beta}$, we need only find some E that matches that syndrome and apply $E^\dagger$.

We can now restate our notion of the distance of a QECC in the language of stabilizers. As before the weight of an error $E \in G_n$ is the number of non-identity terms. The distance of stabilizer code $C(S)$ is the minimum weight element in $N(S) - S$. The code is said to be a $[n, k, d]$ - code.

A distance $2t+1$ code can correct errors on up to t qubits (because $E_{j'}^\dagger E_j \notin N(S) - S$)

wt $\leq 2t$. (t from each error).

Now we turn to the task of finding a basis of $C(S)$. Again, we will define each state by the set of operators which stabilize it. We will do this systematically by finding a set of operations $\bar{Z}_1 \cdots \bar{Z}_k$ such that

$g_1, \ldots, g_{n-k}, \bar{Z}_1, \ldots, \bar{Z}_k$ form an independent + commuting set.

↳ We will elaborate more on how to systematically select the $\bar{Z}_k$'s.

The space V_S is specified by determining a set of k logical qubits chosen so that $\bar{Z}_j$ is the Z operator on the j^{th} logical qubit. The basis state $|\bar{X}_1\ \bar{X}_2 \cdots \bar{X}_k\rangle$ is stabilized by the set $\langle g_1, \ldots, g_\ell, (-1)^{x_1}\bar{Z}_1, \ldots, (-1)^{x_k}\bar{Z}_k\rangle$.

↳ note that Z stabilizes $|0\rangle$ and $-Z$ stabilizes $|1\rangle$.

We can also define logical $\bar{X}$ operators. $\bar{X}_k$ is the unique operator such that

$$\bar{X}_k\ \bar{Z}_k\ \bar{X}_k^\dagger = -\bar{Z}_k$$

and $\bar{X}_k$ commutes with all the g's and all the other $\bar{Z}_j$'s. $(j \neq k)$.

Shor Code Revisited:

The 9-bit Shor code has stabilizer $M_1 \cdots M_8$ (given last lecture)

$M_1 = ZZI\ III\ III$
$M_2 = ZIZ\ III\ III$
$M_3 = III\ ZZI\ III$
$M_4 = III\ ZIZ\ III$
$M_5 = III\ III\ ZZI$
$M_6 = III\ III\ ZIZ$
$M_7 = XXX\ XXX\ III$
$M_8 = XXX\ III\ XXX$

The operator $\bar{Z} = X_1X_2 \cdots X_9$ commutes with all of $M_1 - M_8$ and is independent from $M_1 - M_8$

$|\bar{0}\rangle$ is the state that is stabilized by all of $M_1 - M_8$ and $\bar{Z}$

$|\bar{1}\rangle$ is the state that is stabilized by all of $M_1 - M_8$ and $-\bar{Z}$.

What about a new example that we can derive through the Stabilizer formalism? Also, is it possible to encode a single qubit more efficiently and correct any 1-qubit error?

Consider the Stabilizer generated by the following set of operations on five qubits:

$M_1 = XZZXI$
$M_2 = IXZZX$
$M_3 = XIXZZ$
$M_4 = ZXIXZ$.

This set is independent, commuting and $(M_i)^2 = I \quad \forall\ i = 1,2,3,4$.

Also, any operator in G_5 of weight 1 or 2 fails to commute with at least one of $M_1 - M_4$.

$\Rightarrow$ this code has distance ≥ 3 and can correct all 1 qubit errors.

The operator $YZYII$ does commute with M_1-M_4 and is not in S, so the distance is exactly 3.

→ For example error syndrome for $IYIII$ is $-1\ -1\ 1\ -1$

What are the code words?

Define $\bar{Z} = ZZZZZ$ and $\bar{X} = XXXXX$

Note that $M_1, M_2, M_3, M_4, \bar{Z}$ are mutually commuting and independent.

What are the code words? Define $\tilde{M} = \frac{1}{\sqrt{|S|}} \sum_{M \in S} M$

Note that $\tilde{M} M_i = M_i \tilde{M} = \tilde{M}$

(multiplication by M_i just permutes the elements of S)

Define: $|\bar{0}\rangle = \tilde{M}|00000\rangle$

$|\bar{1}\rangle = \bar{X}|\bar{0}\rangle = \bar{X}\tilde{M}|00000\rangle = \tilde{M}\bar{X}|00000\rangle = \tilde{M}|11111\rangle$

M_i Stabilizes $|\bar{0}\rangle, |\bar{1}\rangle$ $M_i\tilde{M}|00000\rangle = \tilde{M}|0000\rangle = |\bar{0}\rangle$

$M_i\tilde{M}|11111\rangle = \tilde{M}|1111\rangle = |\bar{1}\rangle$

$\bar{Z}$ Stabilizes $|\bar{0}\rangle$ = $\bar{Z}\tilde{M}|00000\rangle = \tilde{M}\bar{Z}|0000\rangle = \tilde{M}|00000\rangle$

$-\bar{Z}$ Stabilizes $|\bar{1}\rangle$ = $-\bar{Z}\tilde{M}|11111\rangle = -\tilde{M}\bar{Z}|1111\rangle = \tilde{M}|11111\rangle$

Now how do we find the $\bar{Z}$ operators systematically?
First we put the check matrix into a standard form:

Notice that we can perform Gaussian elimination on the check matrix and resulting stabilizer remains the same:

Swapping rows: reordering of generators

Swapping columns: reordering qubits

Adding rows: $r_i \leftarrow r_i + r_j \iff g_i \leftarrow g_i g_j$ ← this generates the same group.

So first do Gaussian elimination on X side to get:

$$\begin{array}{c} \\ r \\ n-r \end{array} \begin{array}{c} \overbrace{}^{r}\ \overbrace{}^{n-r} \\ \left[\begin{array}{cc|cc} I & A & B & C \\ 0 & 0 & D & E \end{array}\right] \end{array}$$

Then do Gaussian elimination on E:

$$\begin{array}{c} \\ r \\ n-k-r-s \\ s \end{array} \begin{array}{c} \overbrace{}^{r}\ \overbrace{}^{n-k-r-s}\ \overbrace{}^{s} \\ \left[\begin{array}{ccc|ccc} I & A_1 & A_2 & B & C_1 & C_2 \\ 0 & 0 & 0 & D_1 & I & E_2 \\ 0 & 0 & 0 & D_2 & 0 & 0 \end{array}\right] \end{array}$$

In order for the last s rows to commute with the first r rows, $D_2 = 0$.

$\Rightarrow s = 0$.

Also, can eliminate C_1, using the middle set of rows.

So we have:

$$\begin{array}{c} \\ r \\ n-k-r \end{array}\!\!\begin{array}{c} \begin{array}{ccc} r & n-k-r & k \end{array} \\ \left[\begin{array}{ccc|ccc} I & A_1 & A_2 & B & 0 & C \\ 0 & 0 & 0 & D & I & E \end{array}\right] \end{array} = C.$$

We need to pick $G_z = \begin{array}{c} \begin{array}{ccc|ccc} r & n-k-r & k & r & n-k-r & k \end{array} \\ \left[\begin{array}{ccc|ccc} F_1 & F_2 & F_3 & F_4 & F_5 & F_6 \end{array}\right] \end{array} k$

that commute with the check matrix and are independent (when combined with the stabilizer).

Choose $G_z = [0\,0\,0 \mid A_2^T\ 0\ I]$

$$C \curlywedge G_z = C \begin{array}{c} \begin{array}{c} k \end{array} \\ \left[\begin{array}{c} A_2 \\ 0 \\ I \\ 0 \\ 0 \\ 0 \end{array}\right] \begin{array}{l} r \\ n-k-r \\ k \\ r \\ n-k-r \\ k \end{array} \end{array}$$

Note $A_2 + A_2 = 0$.

To see independence: $\begin{array}{c} \\ r \\ n-r-k \\ k \end{array}\!\!\begin{array}{c} \\ \left[\begin{array}{ccc|ccc} I & A_1 & A_2 & B & 0 & C \\ 0 & 0 & 0 & D & I & E \\ 0 & 0 & 0 & A_2^T & 0 & I \end{array}\right] \end{array}$

Suppose $\sum a_i r_i = 0$.

$a_i = 0 \quad \forall\, i \in [1, \ldots, r]$

because of the I on top.

$a_i = 0 \quad \forall\, i \in [r+1, \ldots, n-k]$

because of the I in middle rows

$a_i = 0 \quad \forall\, i \in [n-k+1, \ldots, n]$

because of the I at the bottom.

Now what about encoding + decoding?

We will discuss how to encode the state $|0\cdots0\rangle$ (k zeros)

In general, one might want to encode an arbitrary k-qubit state $|\psi\rangle$. However for computation, it is sufficient to encode $|0\rangle^k$ since that is the starting state. It's possible to implement gates on the encoded states so decoding only needs to be done at the end. It's the convention to start a quantum circuit with all 0's.

(This is part of Fault-Tolerant quantum comp. which we did not cover.)

To encode $|0\rangle^k$ start with $|0\rangle^n$ and measure each operator $g_1 \cdots g_{n-k}\, \bar{Z}_1 \cdots \bar{Z}_k$. We will make use of the lemma from last time that says

$$\forall\ g \in \{g_1,\ldots,g_{n-k},\bar{Z}_1,\ldots,\bar{Z}_k\} \quad \exists\ h \in G_n \text{ s.t.}$$

$$g h g^\dagger = -h$$
$$g' h g'^\dagger = h \quad \forall\ g' \in \{g_1,\ldots,g_{n-k},\bar{Z}_1,\ldots,\bar{Z}_k\},\ g' \neq g.$$

For each $g \in \{g_1,\ldots,g_{n-k},\bar{Z}_1,\ldots,\bar{Z}_k\}$

if the outcome of the measurement for g is $+1$ $\Rightarrow$ do nothing.

if the outcome is -1, apply h to the state.

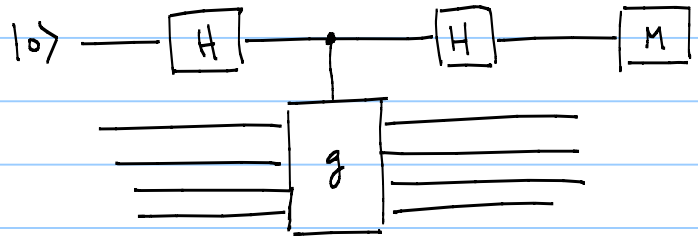
↳ this will have the effect of inverting ($\times(-1)$) the eigenvalue of g but leaving the eigenvalues of the other operators unchanged.

The result is a state whose eigenvalue for each g is $+1$.

It is also possible to perform measurements on the encoded state, so if we are simulating a quantum circuit on encoded data, it's not essential to decode the final state at the end of the circuit. However this is still possible.

The encoding procedure described above is not unitary because it entails measuring the error syndrome. However, there are alternative procedures which are unitary. Decoding can be managed by reversing the unitary encoding circuit.

Meanwhile, everything we have talked about depends on our ability to perform measurements, where the operator corresponding to the measurement is some g in G_n. All the operators in G_n are unitary and Hermitian (meaning the eigenvalues are ± 1). Here is a circuit for such a measurement:



Here is a circuit to measure the operators of the 5-qubit code: