Lecture 7 Benchmarking Pt 2.

Quantum Process Tomography

Analogous to state tomography except instead of learning a density matrix ρ we learn a representation of a quantum process know as the χ matrix.

→ See Mike & Ike 8.4.2 for details.

- Standard QPT requires 2^{n^2} different state tomography experiments. for n qubits
- That is $2^{n^2} \cdot 2^{n^2}$ types of programs to benchmark n qubits.

There are cleverer ways

→ Mohseni et al 2008

→ Gate set tomography pygsti python package

→ QPT w/ compressed sensing (Flammia et al 2012)

scales as $\mathcal{O}\left(\left(\frac{rd}{\varepsilon}\right)^2 \log(d)\right)$ $d = 2^n$

r is rank of the ρ i.e. the dimension of the space spanned by its column vectors.

This is close to optimal

Randomized Benchmarking

Fully charecterizing all the quantum operations of our system is resource intensive. Instead we would like a general metric that tells us how our operations perform on average (for some definition of average)

Defn A group G is a set equipped w/ a binary operation that is:

- closed : $g_1 \cdot g_2 \in G$
- associative : $(g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$
- has an identity : $1_G \cdot g = g$
- is invertible : $\forall g \in G$ there is a g^{-1} s.t $g \cdot g^{-1} = 1_G$

Ex The Pauli group $P_1 \equiv \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}$ w/ matrix multiplication

$XZ = iY$

$p = p^{-1}$ for $p \in \{X, Y, Z\}$

The Pauli group is generated by $\langle X, Y, Z \rangle$ which means all the elements of P_1 can be written as a finite combination of X, Y or Z.

Defn The normalizer of the Pauli group is called the Clifford Group

$$C_1 = \{ U \mid \sigma \in P_1 \Rightarrow U \sigma U^\dagger \in P_1 \}$$

For example $H \in C_1$ as $HZH = X$ etc.

Matrices in C_1 correspond to permutations of the axes of the Bloch sphere. For example it could send $+x$ to $-z$. There are six ways to send the first axis, then four for the next. Thus $|C_1| = 24$ elements.

C_1 is generated by $\underbrace{\{I, RX(\pm\pi), RX(\pm\pi/2), RY(\pm\pi), RY(\pm\pi/2)\}}_{RB_{1Q}}$ (9 elements)

call this the 1 Qubit randomized benchmarking gate set.

$C_1 = \langle RB_{1Q} \rangle$

As C_1 is a group we can define that, for a set of elements $c_1, \ldots, c_n$ each of which is a member of C_1, their completion $A = (c_1 \cdots c_n)^{-1}$. Thus

$$A(c_1 \cdots c_n) = 1_{C_1} = \mathbb{1} \text{ the identity matrix.}$$

Fact for any $c_1 \ldots c_n$ their completion A can always be written as a combination of 3 elements of RB_{1Q}

[single qubit randomized benchmarking]

(1) Choose a series of lengths $l_1 < l_2 \ldots < l_m$

(2) For each l_i
Generate a random sequence of $l_i - 3$ gates taken from RB_{1Q}.

Calculate their completion A and its representation as a combination of 3 elements of RB_{1Q}.

This produces a sequence of l_i elements of RB_{1Q} that compose to the identity operation.

(3) Run this sequence of gates on your qubit and measure if you remain in $|0\rangle$ state

Do this N_e times

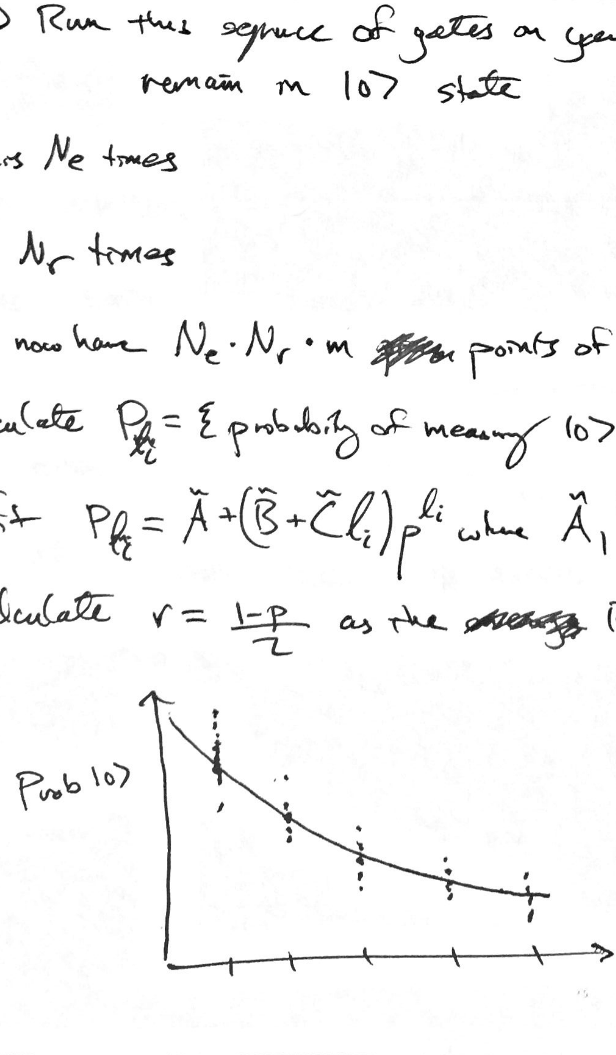
Do this N_r times

You now have $N_e \cdot N_r \cdot m$ ~~…~~ points of data.

(4) Calculate $P_{l_i} = \{$probability of measuring $|0\rangle$ at sequence length $l_i\}$

(5) Fit $P_{l_i} = \breve{A} + (\breve{B} + \breve{C} l_i) p^{l_i}$ where $\breve{A}, \breve{B}, \breve{C}$ and p are fit params.

(6) Calculate $r = \frac{1-p}{2}$ as the ~~…~~ RB number.



Facts about RB (part A)

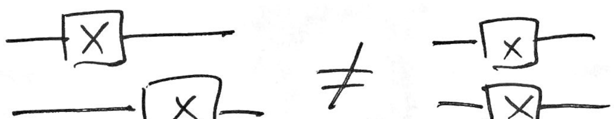
- Gate errors are independent e.g. $\underset{\text{actual noisy gate}}{C_i'} = \Lambda \underset{\text{perfect target gate}}{C_i}$
  then r = avg gate infidelity (Magesan et al 2012)
- When gate errors are dependent, i.e. $C_i' = \Lambda_i C_i$
  then we can use a slightly different procedure (Proctor et al. 2017)
- RB is invariant to SPAM (state preparation and measurement) errors

- The Clifford group is special. It is a so-called unitary 2-design which means that the effect of the gates on the existing error channel is to map it into a depolarizing channel w/ the avg infidelity invariant (Magesan et al 2012)

[Simultaneous Randomized Benchmarking]

~~[illegible]~~ One could apply the ~~[illegible]~~ 1Q RB protocol to each of your available qubits one at a time to get fidelity numbers. However this ignores the interaction between qubits. e.g.

Serial X



parallel X

In <u>simultaneous RB</u> we take data for RB sequences on ~~[illegible]~~ subsets of qubits at the same time.

[Multi-qubit RB]

The protocol generalizes to multiple qubit in a way that is mathematically straightforward but that can be ~~hard~~ hard to implement in practice.

Defn
The Pauli group on N-qubits is gen by

$$P_N = \{p_1 \otimes \dots \otimes p_N \text{ for } p_i \in P_1\}$$

The n-qubit Clifford group is then the normalizer for P_N as

$$C_N = \{U \mid p \in P_N \Rightarrow U p U^\dagger \in P_N\}$$

The rest of the protocol follows through the same way.

Ex1 2Q RB has ~~the~~ a following generating set:

$$RB_{2Q} = \{p_1 \otimes p_2 \text{ for } p_1, p_2 \in P_1\} \cup \{CZ\}$$

s.t. $C_2 = \langle RB_{2Q} \rangle$

Ex2 $C_2 = \langle \{ (NOT, H), \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix} = S \} \rangle$

However, algorithms for doing the generation of random sequences that compose to the identity become harder as the size of the Clifford group grows quickly

n	1	2	3	4	5
$\lvert C_n \rvert$	24	11,520	92,897,280	12,128,668,876,800	25 quadrillion

[Interleaved RB] (Magesan 2014)

RB gives us a number that benchmarks the overall control of a set of operations. However we can modify it to learn about the error of a particular gate.

(1) Do normal RB and calculate p

(2) Repeat RB but ~~place~~ insert the target gate C_a as the first element in the sequence and in between every other gate. Calculate p_{C_a}

(3) $$r_a^{est} = \frac{(d-1)(1 - p_a/p)}{d} \qquad d = 2^n \leftarrow \# \text{ of qubits}$$

<u>Thm</u> [Gottesman-Knill] (Gottesman 1998, M&I ~~10.5.4~~ 10.5)

~~xxxxx~~ Quantum programs that consist only of Clifford operations followed by one round of measurements can be efficiently simulated by classical computers

So does this mean RB can only be used on the "boring" classically simulable operations?

<u>No</u> and this is an active research area.

$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/8} \end{pmatrix}$

- Onorati et al 2018 for T gates (Clifford + T is universal)

  França & Hashagen 2018 " " " "

Benchmarking - Contextuality

So far our benchmarking thus far we have been concerned w/ the fidelity of particular operations or sets of operations.

We may also be ~~[illegible]~~ interested in more global properties such as if our processor is doing something that is clearly quantum.

In these next two sections we will focus on benchmarks that to some extent can be considered benchmarks of the "quantumness" of your QPU.

Defn Contextual (informal)

A physical system is <u>contextual</u> iff the state of the system cannot be defined without including the choice of observables as parameters.

Thus the systems state cannot be defined w/o the measurement context.

Toy example

A ball-cup system has two observables, each of which has two states.

- You can peek at the ball hidden under the cup and see if it is black or white
- You can weigh the cup and see if it is heavier or lighter than 0.5 lbs

How do you fill in the following table

weigh, weigh :	heavy, light
weigh, peek :	heavy, white
peek, weigh :	white, light
peek, peek :	? ?

For a non contextual theory this must be white, white.

But! In quantum and other ~~[illegible]~~ contextual theories we may get black/black!

(Hardy 1993)

Lets look at a famous example due to Mermin.
We will find that the quantum protocol produces an outcome that would be impossible for a non-contextual theory to produce.

Take a 3-qubit system and perform the following sets of measurements.

get one output number by XOR on outputs.

$$c = X_2 X_1 X_0 \qquad a_1 = Y_2 Y_1 X_0$$
$$a_2 = X_2 Y_1 Y_0$$
$$a_3 = Y_2 X_1 Y_0$$

calculate $c \stackrel{?}{=} a_1 \oplus a_2 \oplus a_3 = a$

Imagine if $\{X, Y\}_i$ had fixed non-contextual values. e.g. $X_i = 0$, $Y_i = 1$

$$c = 0 \oplus 0 \oplus 0 = 0 \qquad a_1 = 1 \oplus 1 \oplus 0 = 0$$
$$a_2 = 0 \oplus 1 \oplus 1 = 0 \Rightarrow a = 0 \qquad a == c$$
$$a_3 = 1 \oplus 0 \oplus 1 = 0$$

In fact for all (non-contextual) choices of x_i and y_i $a == c$

$$c = x_2 \oplus x_1 \oplus x_0 \qquad a = \underbrace{\cancel{y_2} \oplus \cancel{y_1} \oplus x_0}_{a_1} \oplus \underbrace{x_2 \oplus \cancel{y_1} \oplus \cancel{y_0}}_{a_2} \oplus \underbrace{\cancel{y_2} \oplus x_1 \oplus \cancel{y_0}}_{a_3}$$

$$c == a = x_0 \oplus x_2 \oplus x_1$$

Now lets imagine that our system is quantum and is in the

$$\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \text{ state. (called a GHZ state)}$$

* exercise to the reader to check algebra

We find that $\langle XXX \rangle = -\langle YYX \rangle \langle XYY \rangle \langle YXY \rangle$

$$c \neq a$$

whoa!

Thus if we make this state and make these measurements we will find an outcome that is impossible for non-contextual theory

To make this into a benchmark ~~we~~ it is not enough to observe one outcome of $c \neq a$.

Since the system is noisy we may have seen an error rather than real evidence of contextuality. Thus we must calculate the probability that $c \neq a$ on our QPU compared to a contextual theory w/ some random bit flip % chance at measurement.

This is a so-called ~~(?)~~ $(3, 2, 2)$ ~~(?)~~ Mermin or all vs nothing benchmark.

For - 3 systems
- each w/ 2 observables $\{X, Y\}$
- and each observable has two outcomes $\{\pm 1\}$

This can be generalized to (N, M, D) benchmarks see.
Gogioso and Zeng 2017

While contextuality benchmarks tell us about the "quantumness" of states produced on our QPU, they can be fooled. It is not hard to write a classical simulation of a contextual theory that will pass ~~our~~ benchmark.

Thus we are interested in benchmarks that separate the QPU's performance from anything that could be done classically.

Quantum Supremacy

Definition The milestone of quantum supremacy consists of:

(1) A mathematical proof that a given problem has a superpolynomial separation between quantum and classical (at least up to widely accepted theoretical assumptions)

(2) The exhibition of the solution of this problem by a quantum computer at a performance (size, speed, or efficiency) that is infeasible w/ any classical computer in existence

Defn Weak quantum supremacy: just (2) of quantum supremacy

Defn Quantum Advantage: (2) but for a valuable commercial problem.

Defn Strong Quantum Advantage: Quantum advantage w/ (1) or equivalently quantum supremacy for a valuable problem

	Practical	Foundational
Any Problem	WQS	QS
Valuable problem	QA	SQA

Quantum Supremacy through Porter-Thomas Sampling (Boixo et al 2016)

One of the most widely known approaches to proving near-term quantum supremacy is through Porter Thomas sampling.
In essence, ~~the~~ the output of random quantum programs is not only distinguishable, but is hard to simulate w/ classical computation.

Classical

Imagine ~~a~~ random boolean functions $0^{\otimes n} \xrightarrow{f} \{0,1\}^{\otimes n}$
The distribution over output bitstrings is uniform, i.e. $p = \{p_i\}_{i \in \{0,1\}^n}$ has $p_i = \frac{1}{2^n}$
The meta-distribution (a histogram of the p_i values) is a delta function as all p_i are the same.

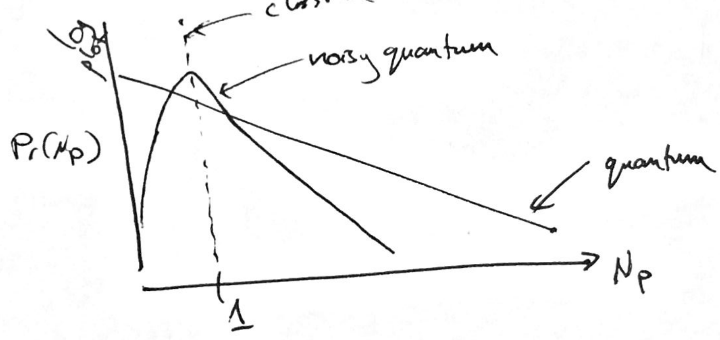
Quantum

$|0\rangle^{\otimes n} \xrightarrow[\text{+ measure}]{U^t} \{0,1\}^{\otimes n}$ looks different.
(U^t: random unitaries)

The meta-distribution is

$$P_r(N_p) = e^{-N_p}$$

we rescale the p_i by $N = 2^n$



corresponds to $N_p = 1 \Rightarrow p_i = \frac{1}{2^n}$ or the classical case

We compare quantum to classical by using the crossentropy difference.

Defn For discrete distributions the crossentropy $H(p,q) = -\sum_{x \in X} p(x) \log(q(x))$

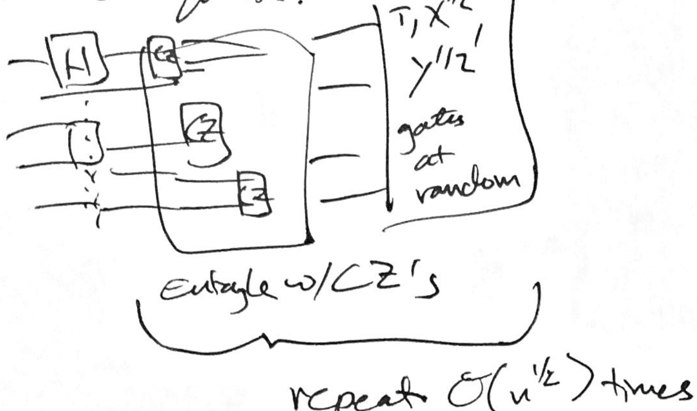
Defn Cross-entropy difference is relative to a reference.
Let P_T be the porter thomas distribution
δ be the delta meta-distribution that comes when $p_i = \frac{1}{2^n}$

Then $\Delta H(p_A) = H(\delta, P_T) - H(p_A, P_T)$ (measured)

$$= \sum_j \left(\frac{1}{N} - p_A(x_j|U)\right) \log \frac{1}{P_T(x_j)}$$

It turns out that nearest neighbor circuits can approximate a random U well enough for this to work. (Aaronson & Chen 2017)

Google Proposal is the following to generate U on a square grid of qubits:



7×7 array $= 49$

40 layers

Results so far!

9 qubits yes

TBD on more.