Benchmarking - Contextuality

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

We may also be 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 contextual 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 noncontextual theory this must be white, white.

But! In quantum and other 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.

$c = X_2 X_1 X_0$ $\quad a_1 = Y_2 X_1 X_0$

$\quad a_2 = X_2 Y_1 X_0$

$\quad a_3 = Y_2 X_1 Y_0$

get one output number by XOR on outputs.

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

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

$c = 0 \oplus 0 \oplus 0 = 0$ $\quad a_1 = 1 \oplus 1 \oplus 0 = 0$

$\quad a_2 = 0 \oplus 1 \oplus 1 = 0 \Rightarrow a = 0$ $\quad a == c$

$\quad a_3 = 1 \oplus 0 \oplus 1 = 0$

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

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

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

Now lets imagine that our system is quantum and is the

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

* exercise to the reader to check algebra

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

whoa! $\quad c \neq a$

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 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 out 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

Definiton 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 existance

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 equally well 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 output of random quantum programs is not only distinguishable, but is hard to simulate w/ classical computation.

Classical

Imagine 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

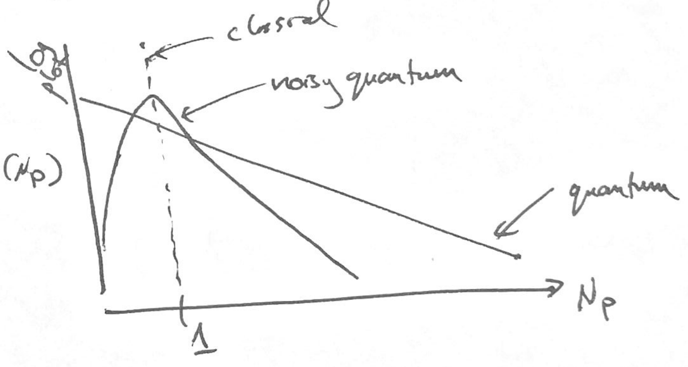
$$|0\rangle^{\otimes n} \xrightarrow[+\text{ Measure}]{U^t} \{0,1\}^{\otimes n}$$ random unitaries

Looks different.

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

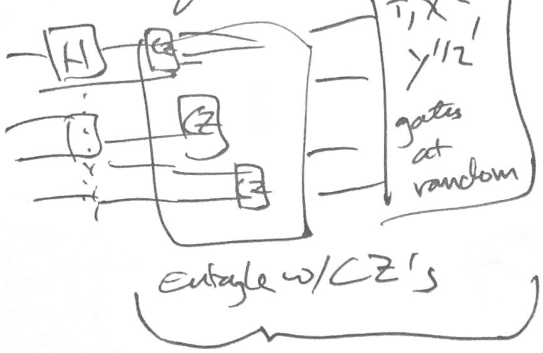
δ be the delta meta-distribution that comes when $p_i = \frac{1}{2^n}$

Then $\Delta H(\overset{\text{measure}}{p_A}) = H(\delta, P_T) - H(p_A, P_T)$

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

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:



repeat $O(n^{1/2})$ times

7x7 array = 49

40 layers

Results so far!

9 qubits yes (Neill et al 2017)

TBD on more.

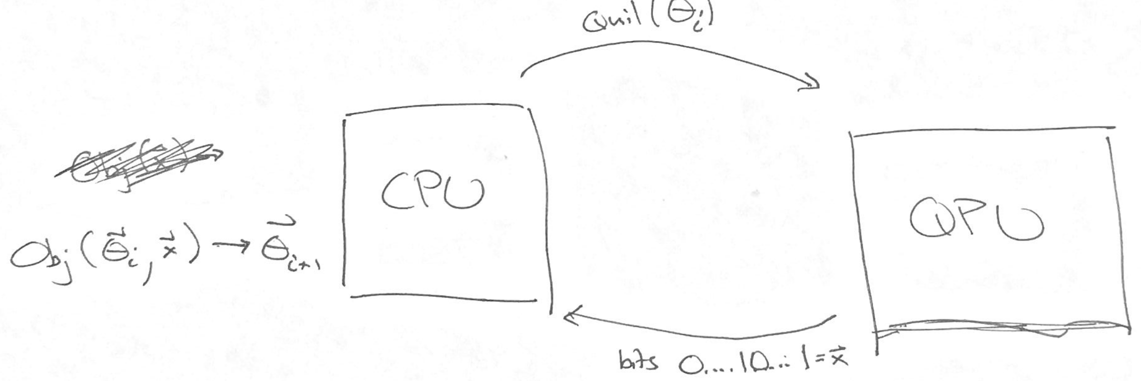
The Varrational Quantum Eigensolver

In the last few lectures we have shown that noise is a critical part of today's quantum programming. It will remain critial for many years even as hardware continues to improve. This means there is a need for programming models that can be robust to QPU errors.

VQE (the variational quantum eigensolver) is a key example of how this can work.

VQE is an example of so-called hybrid quantum/classical or variational quantum algorithms.

In the variational paradigm, the parameters of a quantum program are learned with the following feedback loop:


$\text{Quil}(\vec{\theta}_i)$
CPU
QPU
$\text{Obj}(\vec{\theta}_i, \vec{x}) \rightarrow \vec{\theta}_{i+1}$
bits $0 \ldots 1 0 \ldots 1 = \vec{x}$

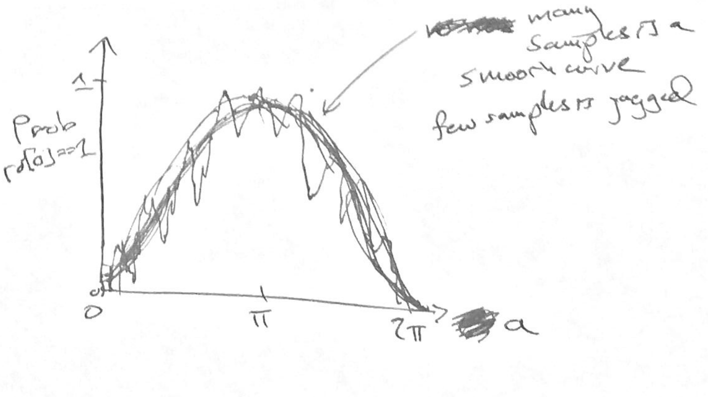
where we use an objective function to update parameters used to generate a Quil program.

Ex (notebook)

Let Quil $(\vec{\theta}_i)$ be def ansatz(a:real):
return Program(RX(a, 0)).measure_all()

Objective function is probability that ro[0] == 1

Optimizing gives this landscape



max of obj gives $a = \pi$

Note that this is a nontrivial optimization for several reasons.

- we have "sampling error" where we may not have run enough trials
- we are optimizing over a noisy landscape due to quantum noise.

Thus variational optimization for quantum programming is an optimization of a parameterized distribution over a noisy landscape.

As we will see throughout the class, this method can be used to make noise robust quantum algorithms for everything from classifiers to quantum chemistry simulations.

However there is no free lunch. While variational algorithms are more noise robust it is often an open research question as to how robust they are and/or what speedup they may give. Thus we trade proven asymptotic speedups (from Shor's alg etc.) for noise robustness.

VQE

Shows how we can use variational quantum programming to find the smallest eigenvalues and eigenvectors of large matrices, in fact some can be exponentially large!

Defn An EIGEN problem is given a representation of a matrix A find λ_0 and $\vec{x}_0$ the smallest eigenvalue (and corresponding eigenvector) of A.

Thm [The variational principle]

Let H be a $2^N \times 2^N$ Hermitian complex matrix w/ eigenvalues $\{\lambda_i\}$ (ordered by size from smallest to largest) and eigenvectors $\{\vec{x}_i\}$. Then for any vector $|y\rangle$ (complex and size 2^N)

$$\frac{\langle y|H|y\rangle}{\langle y|y\rangle} \geq \lambda_0$$

AND

Equality of the above ~~equation~~ holds only when $|y\rangle = |\vec{x}_0\rangle$

Proof (of variational principle)

As H is Hermitian the eigenvectors are orthogonal and so we can expand $|y\rangle$ in the eigenstates of H:

complex coefficients

$$|y\rangle = \sum_i c_i |x_i\rangle$$

We then have $\langle y|y\rangle = \sum_{ij} c_i^* c_j \langle x_i|x_j\rangle = \sum_i |c_i|^2$ due to orthogonality of $|x_i\rangle$

and $\langle y|H|y\rangle = \sum_{ij} c_i^* c_j \langle x_i|H|x_j\rangle = \sum_{ij} c_i^* c_j \lambda_i \delta_{ij}$ by orthog and def of eigenvectors/values

$$= \sum_i \lambda_i |c_i|^2$$

We then have $$\frac{\langle y|H|y\rangle}{\langle y|y\rangle} = \frac{\sum_i \lambda_i |c_i|^2}{\sum_i |c_i|^2} \geq \frac{\lambda_0 \sum_i |c_i|^2}{\sum_i |c_i|^2} = \lambda_0 \qquad \square$$

Since $|c_i|^2 \geq 0$ and $\lambda_n \geq \lambda_0$ we have that

This gives us a method for VQE.

Alg VQE (1) Let Quil$(\vec{\theta})$ be a quantum program that produces some state $|y(\vec{\theta})\rangle$

(2) Run the program and measure the expectation value $\langle y(\vec{\theta}_i)|H|y(\vec{\theta}_i)\rangle$

(3) As $|y(\vec{\theta})\rangle$ is a normalized quantum state we know this expectation value is $\geq \lambda_0$. Thus update θ_{i+1} to minimize $\langle y(\theta)|H|y(\theta)\rangle$

(4) This will then approach λ_0 from above.

Q: Doesn't this only work for Hermitian matrices?

It is easy to generalize as for any square matrix A ~~$\begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix}$~~ $C = \begin{pmatrix} 0 & A \\ A^{+} & 0 \end{pmatrix}$ is Hermitian.

C also shares eigenvalues as $C\begin{pmatrix} 0 \\ x_i \end{pmatrix} = \lambda_i \begin{pmatrix} 0 \\ x_i \end{pmatrix}$

Q: How to calculate $\langle y|H|y\rangle$? (when w/ Quil we can only measure the Z observable?)

A: we need a decomposition of $H = \sum_i h_i \prod_j \sigma$ where $\sigma \in P_N$ for H of size $2^n \times 2^n$

Ex Let $H = Z_0 + X_1 Z_0$

→ First run and measure statistics on qubit 0 MEASURE 0 ro[0]

→ Then run and append RY(-π/2) 1 ← convert Z meas to an X meas

MEASURE 1 ro[1]

MEASURE 0 ro[0]

$$\langle y|H|y\rangle = \langle y|Z_0|y\rangle + \langle y|X_1 Z_0|y\rangle$$

More details on how to implement this next time!