CPEN 400Q / EECE 571Q Lecture 18
Error mitigation, and introducing QAOA

Thursday 17 March 2022

Announcements

- Assignment 4 coming (will be due at end of term)

Last time

We learned how to use trace distance (T) and fidelity (F) to compare two density matrices:

Last time

We discussed some ways by which quantum computers are compared.

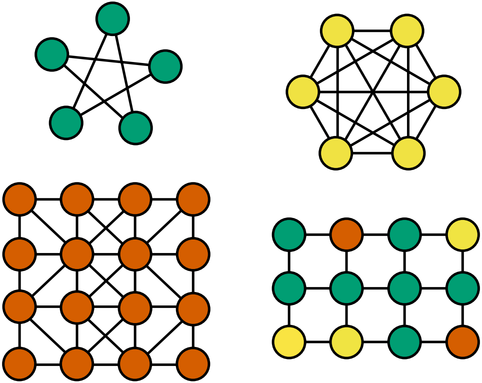
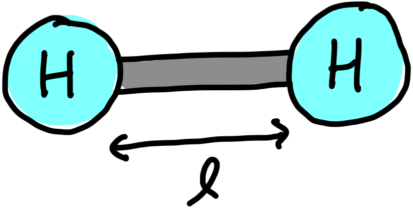


Image credit: quantum volume demo, `https://pennylane.ai/qml/demos/quantum_volume.html`

Last time

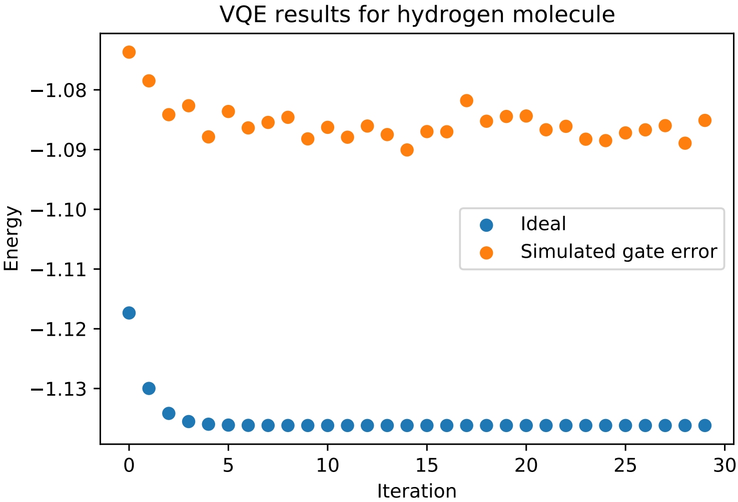
We used VQE to solve a small quantum chemistry problem:



The Hamiltonian uses four qubits, and we needed only a single variational parameter to produce its ground state,

Last time

Then we tried running VQE on a simulated noisy device...



Learning outcomes

- Mitigate noise using zero-noise extrapolation
- Perform basic quantum state tomography
- Describe the underlying ideas of adiabatic quantum computation, and the quantum approximate optimization algorithm

Error mitigation

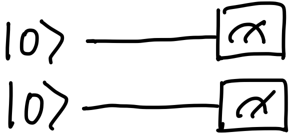
Current quantum hardware is noisy. Noise comes from a variety of sources, and depends on the qubit technology and architecture.

We need to do a combination of:

- Better-characterizing the behaviour of devices to learn how to improve their operation
- Processing the results to mitigate the effects of noise as much as possible

Measurement error mitigation

Errors can occur in the measurement process where states are read out incorrectly.

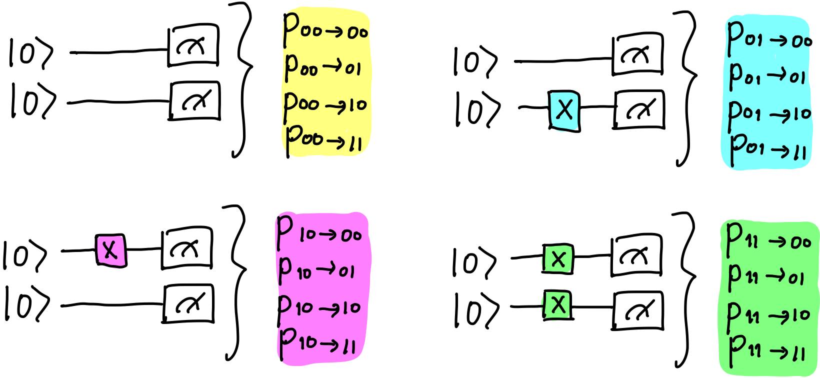


$$
\left.\right\}\quad
\begin{array}{lll}
00: & 0.95 & p_{00\to 00} \\
01: & 0.02 & p_{00\to 01} \\
10: & 0.02 & p_{00\to 10} \\
11: & 0.01 & p_{00\to 11}
\end{array}
$$

These kinds of errors are quite straightforward to mitigate.

Measurement error mitigation

Check what happens with all possible input states:



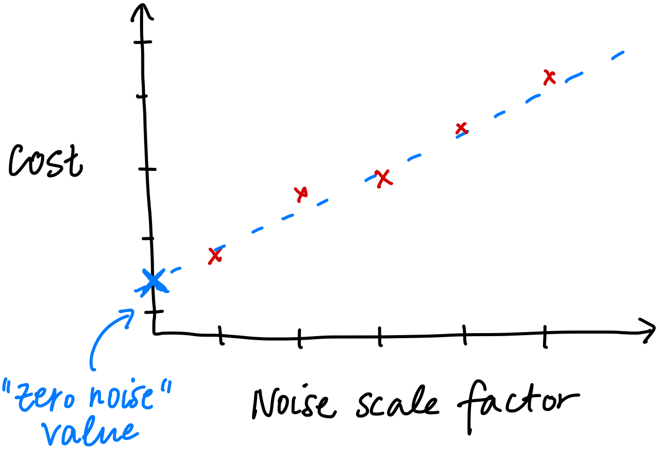
We can put the results from our calibration circuits into a matrix:

$$M = \begin{bmatrix} p_{00\to 00} & p_{01\to 00} & p_{10\to 00} & p_{11\to 00} \\ p_{00\to 01} & p_{01\to 01} & p_{10\to 01} & p_{11\to 01} \\ p_{00\to 10} & p_{01\to 10} & p_{10\to 10} & p_{11\to 10} \\ p_{00\to 11} & p_{01\to 11} & p_{10\to 11} & p_{11\to 11} \end{bmatrix}$$

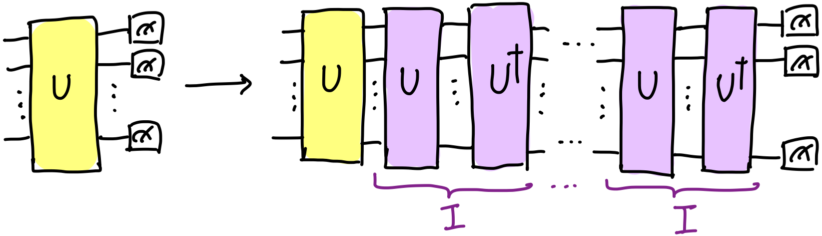
Can suppose that the probability vector P_{noisy} we get at the end of a quantum algorithm is related to the ideal one, P_{ideal}, under multiplication by M since that's what we see:

So to get the ideal results:

Modify the circuits to systematically *increase* scale of the noise, then extrapolate down to the zero-noise limit.



Unitary folding

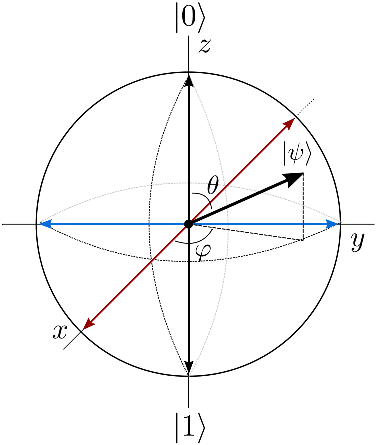


Let's code up a very basic version of this in PennyLane.

More sophisticated version: Python package `mitiq`
`https://github.com/unitaryfund/mitiq`

Quantum state tomography

In order to quantify how close the state obtained from a noisy process is from the true state, we need a way of determining what that state is (i.e., its density matrix). We can reconstruct a state by taking an *informationally complete* set of measurements.



Measure:

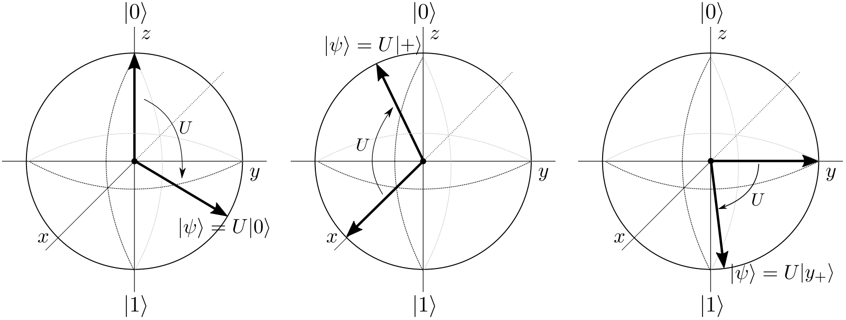
$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Quantum process tomography

Similar method for learning about quantum processes: reconstruct an operation based on how it acts on known states.



(Example: a unitary operation on a single qubit.)

Quantum tomography: MUBs

Tomography is, in some sense, a "solved problem": we know what optimal sets of measurements look like for most cases. (The amount of them scales exponentially in the number of qubits...)

In dimension d, two bases, $A = \{|a_i\rangle\}$ and $B = \{|b_i\rangle\}$ are *mutually unbiased* if for all $|a_i\rangle, |b_j\rangle$,

A complete set of $d + 1$ mutually unbiased bases (**MUBs**) comprises an optimal set of measurement bases.

Quantum tomography: MUBs

Pro: systematic method of construction using finite fields and the Pauli group

Con: complete sets are only known in systems with d prime or power-of-prime.

Example: $d = 2^2$ (2-qubit case). Partition Paulis into $d + 1$ sets of $d - 1$ commuting operators; their shared eigenbases are the MUBs.

Set ID	Paulis		
1	ZZ	IZ	ZI
2	XX	IX	XI
3	YY	IY	YI
4	XY	YZ	ZX
5	YX	XZ	ZY

Quantum tomography: SIC-POVMs

Alternative: symmetric, informationally complete positive operator-valued measures (**SIC-POVMs**).

In dimension d, this is a POVM with d^2 elements $\{\Pi_i\}$[1] and some additional special properties:

- all Π_i are *rank-1 projectors*, i.e., $\Pi_i = |\psi_i\rangle \langle\psi_i|$
- all pairwise inner products are equal:

[1] Recall that for a POVM, $\sum_i \Pi_i = I$.

Pro: unlike MUBs, conjectured to exist in every dimension

Con: no one has been able to *prove* this (if you can do so, you would be famous)

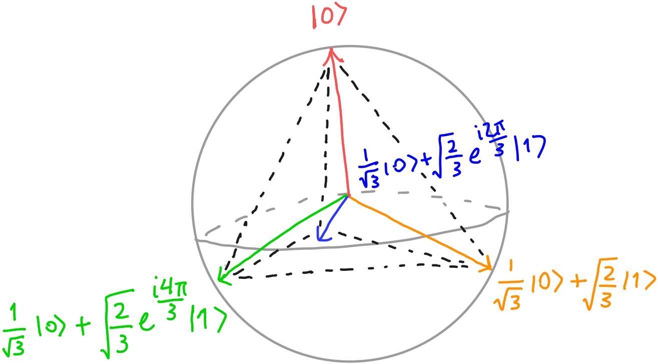


Image credit: `https://pennylane.ai/qml/demos/tutorial_unitary_designs.html`

Let's carry out a very simple state reconstruction on our noisy state.

The d-dimensional Pauli operators are a basis, and are orthogonal w.r.t. the inner product $\mathrm{Tr}(P_i P_j) = d\delta_{ij}$.

Quantum tomography

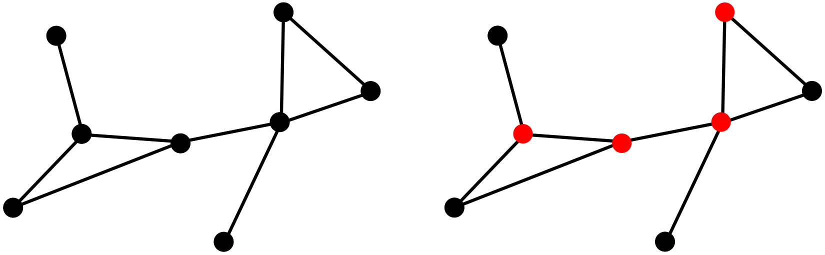
What happens when we multiply ρ by a particular Pauli operator P_j and take the trace?

A simple way to reconstruct the state is to compute the expectation value of every Pauli operator for the final state.

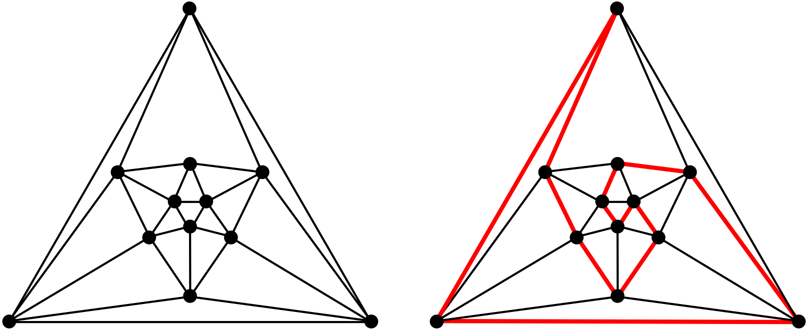
Let's try it.

Quantum approximate optimization algorithm

Example: Given a graph $G = (V, E)$, what is the *smallest number of vertices* you can colour such that every edge in the graph is attached to at least one coloured vertex?



Example: Given a graph, can we find a path through it that visits every node *exactly once* and returns to the starting point?



(In graph theory terms, can we find a *Hamiltonian cycle*?)

Example: You have the opportunity to purchase 100 units of stocks from a fixed list of assets. You know the average returns of each stock, and their covariances.

Stock	Avg. return
AAA	3.44 %
BBB	2.21 %
CCC	-0.28 %
$\vdots$	$\vdots$

Cov.	AAA	BBB	$\cdots$
AAA	0.0038	0.002	$\cdots$
BBB	0.002	-0.006	$\cdots$
CCC	0.014	-0.0008	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$

Suppose you're restricted to buying no more than 5 of any stock.

Which stocks, and how many of each, should you purchase, to maximize your profits?

Adiabatic quantum computing (AQC)

The structure of a classical optimization problem is something like:

$$\min_{\vec{x}} \ \ \mathrm{cost}(\vec{x}) \quad \text{subject to constraints}(\vec{x})$$

where $\vec{x}$ is a multi-dimensional vector of parameters in the problem space.

Adiabatic quantum computing (AQC)

The structure of a classical optimization problem is something like:

$$\min_{\vec{x}} \ \mathrm{cost}(\vec{x}) \quad \text{subject to constraints}(\vec{x})$$

where $\vec{x}$ is a multi-dimensional vector of parameters in the problem space.

In a physical context, optimization can be interpreted as an energy minimization problem.

Optimization	**Physical system**
$\vec{x}$	State of the system
$\mathrm{cost}(\vec{x})$	Hamiltonian
Optimum $\vec{x^*}$	Ground state
$\mathrm{cost}(\vec{x^*})$	Ground state energy

Adiabatic quantum computing (AQC)

Recall that every unitary U is directly related to a Hermitian Hamiltonian H under the correspondence

We know that we can use gate model QC to *simulate* the evolution of a Hamiltonian.

Instead of simulating the Hamiltonians, **adiabatic quantum computing** works with them directly to perform computations. It is generally used to solve **optimization problems**.

Adiabatic quantum computing (AQC)

1. Design a cost Hamiltonian whose ground state represents the solution to our optimization problem
2. Prepare a system in the ground state of an easy-to-prepare mixer Hamiltonian
3. Perform adiabatic evolution to transform the system from the ground state of mixer Hamiltonian to the ground state of the cost Hamiltonian, which is our solution

Why would we want to do this?

Theorem:

"A physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum."

What we can take from this:

If we initialize a system in the lowest energy state and perturb it slowly enough, it will remain in the lowest energy state (with respect to the changed system)

Let H_m be a **mixer Hamiltonian** whose ground state can be easily prepared.

Let H_c be a **cost Hamiltonian** whose ground state represents the solution to a problem of interest.

Adiabatic evolution is expressed mathematically as the function

The parameter s is representative of time; s goes from 0 to 1; $A(s)$ decreases to 0 with time and $B(s)$ increases from 0.

Quantum annealing

D-Wave makes quantum annealers: these are a physical implementation of AQC for a limited set of Hamiltonians.

Image credit:
`www.dwavesys.com/tutorials/background-reading-series/introduction-d-wave-quantum-hardware`

QAOA is a gate-model algorithm that can obtain approximate solutions to combinatorial optimization problems.

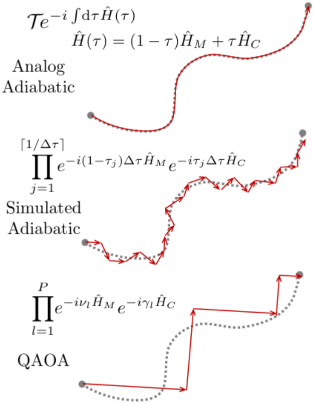
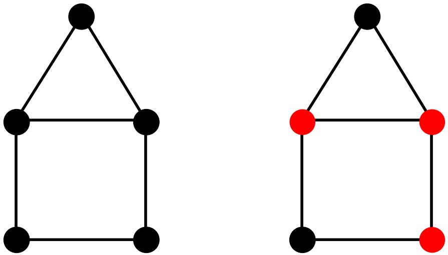


Image credit: G. Verdon, M. Broughton, J. Biamonte. *A quantum algorithm to train neural networks using low-depth circuits.* `https://arxiv.org/abs/1712.05304`

How do we turn an optimization problem for some graph into a Hamiltonian?

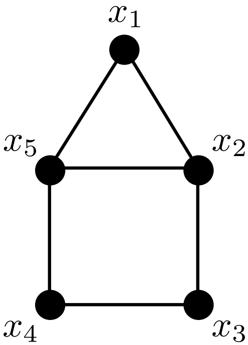
Let's start small, and consider the problem of vertex cover of a graph $G = (V, E)$.



First, we will define a cost function, who minimum cost will correspond to the optimal set of vertices to colour. Then, we will turn it into a Hamiltonian.

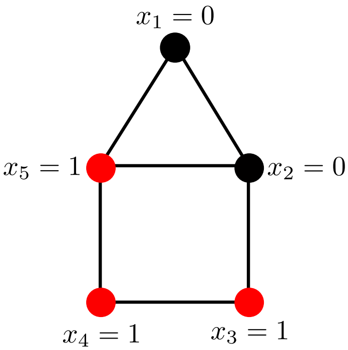
Motivating example: vertex cover

Whether or not a vertex is coloured is a *binary variable*.

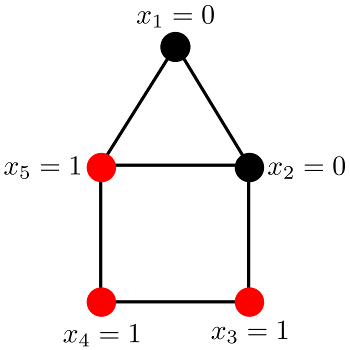


Motivating example: vertex cover

Let's assign coloured vertices to have value 1, and un-coloured 0.



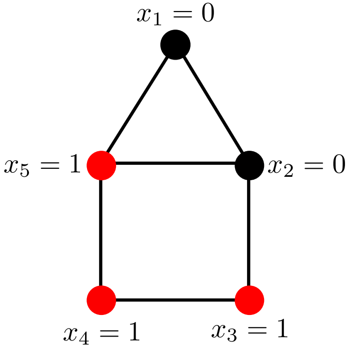
Now that we have our variables, how do we come up with a minimizable cost function that represents the problem?



We need every edge to be next to a coloured vertex. Design a cost function that penalizes edges that are not, but favours ones that are.

Intuitively, find a function of two vertices that is 0 if the colouring is valid, and 1 if it is not.

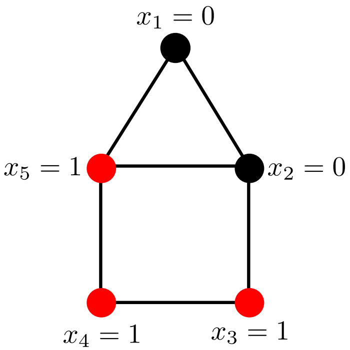
Motivating example: vertex cover



Consider for each edge *ij* the function

The possible values are:

Motivating example: vertex cover

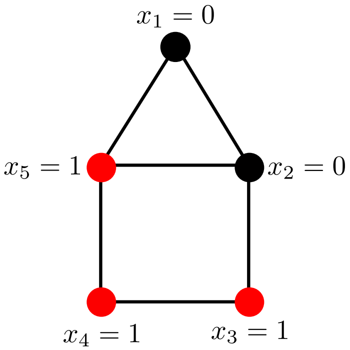


Then in an optimal colouring,

for all edges $ij \in E$.

So we can write

Motivating example: vertex cover



However recall that we also want to colour the fewest vertices. The cost should also depend on the number of coloured vertices.

Solution: add to our cost

The full cost function is then

1. How do we turn this into a Hamiltonian?
2. How do we find its minimum energy / configuration on a quantum computer?

$$\min_{\vec{x}} \left(\sum_{ij \in E} (1 - x_i)(1 - x_j) + \sum_{i \in V} x_i \right)$$

First thing to consider is the problem domain: x_i are binary variables. We have qubits, which can be $|0\rangle$ and $|1\rangle$.

But since we want to turn this into a Hamiltonian and compute a cost (i.e., measure its expectation value), it's more straightforward to map 0 and 1 to *expectation values* associated to $|0\rangle$ and $|1\rangle$.

Usually we consider expectation values of Pauli Z.

We will make the mapping

This associates $x_i = 0$ to $z_i = 1$ (corresponds to $|0\rangle$), and $x_i = 1$ to $z_i = -1$ (corresponds to $|1\rangle$).

Let's expand our cost function and make this substitution.

Hamiltonian translation

Substitute:

Expand:

Collect:

Consider now that: the total number of edges and vertices are constant - they will provide only an “offset” to the cost, and the values of the variables don’t matter.

And finally, the absolute value doesn’t matter, so we can rescale:

Hamiltonian translation

Can also weight the terms differently depending on which constraint is more important (i.e., if you care more about just getting a valid colouring, weight the first one more).

To turn this into a Hamiltonian, recall that

- Each z_i represents an expectation value of Z_i
- Computing expectation values is linear

Next time: we will look at the actual QAOA that can find the optimal configuration / minimum energy.

Next time

Content:

- Continue with QAOA

Action items:

1. Final project

Recommended reading:

- Qiskit tutorial on measurement error mitigation: https://qiskit.org/textbook/ch-quantum-hardware/measurement-error-mitigation.html
- mitiq documentation, for more fun error mitigation: https://mitiq.readthedocs.io/en/stable/guide/guide.html
- New preprint, *Error mitigation increases the effective quantum volume of quantum computers*. https://arxiv.org/abs/2203.05489