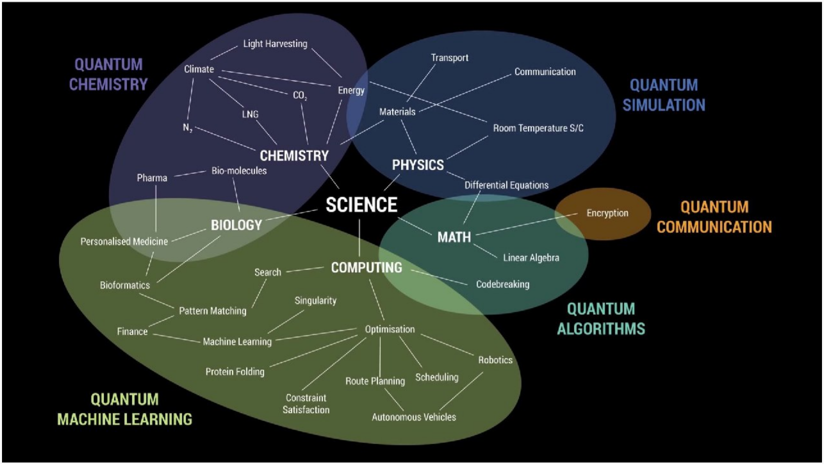
Quantum Algorithm for Principal Component Analysis

Michele Scandelli, Riccardo Santambrogio, Daniele Tagliabue

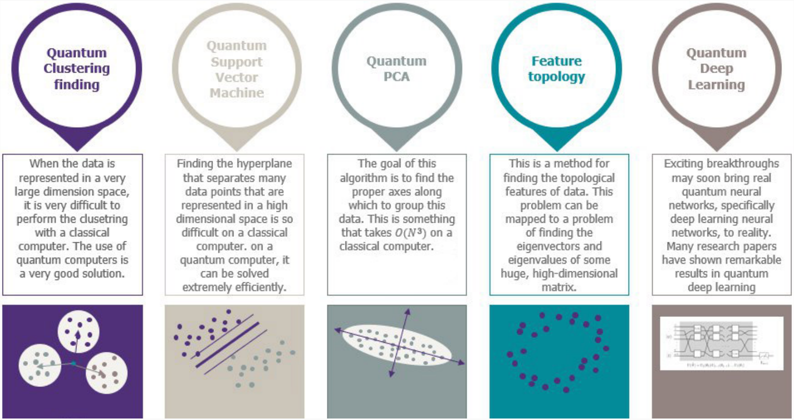
August 2020

Introduction

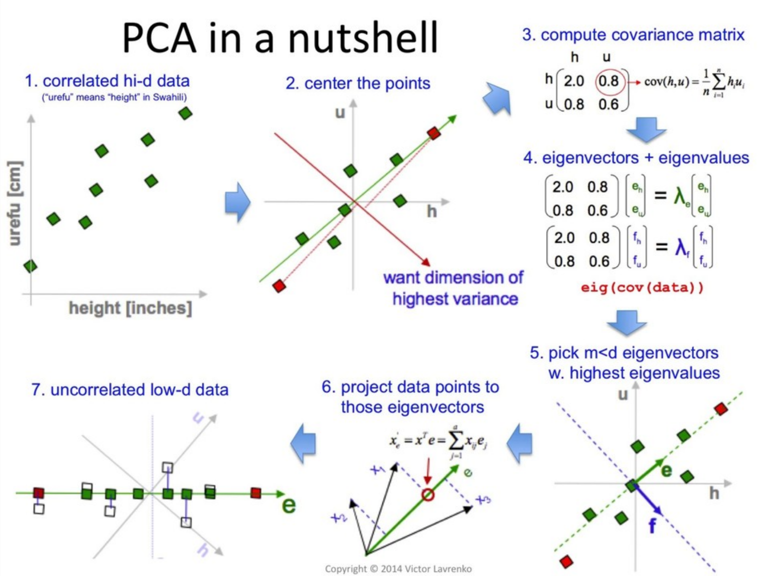
Applications of Quantum Computing



Quantum Machine Learning

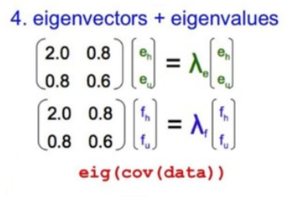


Principal Component Analysis



Complexity of Principal Component Analysis

- Must perform eigendecomposition, which is computationally intensive



- Given this bound, standard implementations of PCA usually run in $O(n^3)$, where n is the number of dimensions (features)
- This is the step we will focus on

The Quantum Algorithm

Quantum PCA

Given a covariance matrix X, normalized w.r.t. its trace, we aim at computing its top-k eigendecomposition by means of a quantum approach

The main technique we will exploit in order to do so is

- Quantum Phase Estimation
  - Phase Kickback
  - Inverse Quantum Fourier Transform

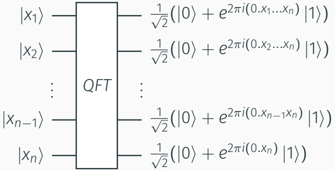
QPE - *Setup*

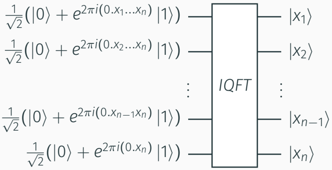
- X is a matrix having eigenvector v and corresponding eigenvalue λ
  i.e. $Xv = \lambda v$
- then v is also eigenvector of e^X, with corresponding eigenvalue e^λ
  i.e. $e^X v = e^\lambda v$

(Proven using Taylor series)

- Consider $U = e^{iX}$ and its eigenvalue $e^{i\lambda} = e^{i2\pi\varphi}$
- Given a quantum state of the form $\frac{1}{\sqrt{2^N}} \sum_{y=0}^{2^N-1} e^{2\pi i \varphi y} |y\rangle$ we can easily determine φ applying the inverse of the *Quantum Fourier Transform*

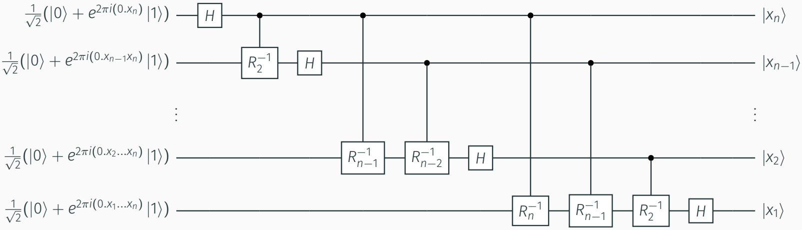
- More specifically, we obtain the best approximation for the binary fraction expansion of φ over n bits
- $\varphi \approx 0.x_1 x_2 \ldots x_{n-1} x_n$





QPE - *Inverse Quantum Fourier Transform*

Circuit for the Inverse QFT



Where $R_k^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{pmatrix}$

- We know how to obtain φ, and therefore also λ, provided we have the quantum state $\frac{1}{\sqrt{2^N}} \sum_{y=0}^{2^N-1} e^{2\pi i \varphi y} |y\rangle$
- But how can we build such a state?
- *Amplitude Encoding* and *Phase Kickback*

- The eigenvector corresponding to $e^{i\lambda}$, for U, is ψ_λ
- Using *Amplitude Encoding* then we could build the eigenstate (*i.e.* quantum state) $|\psi_\lambda\rangle$ over $log_2(|\psi_\lambda|)$ qubits

$$\psi_\lambda = \begin{pmatrix} \psi_{\lambda 0} \\ \psi_{\lambda 1} \\ \vdots \\ \psi_{\lambda n} \end{pmatrix} \implies \psi_{\lambda 0} |0...00\rangle + \psi_{\lambda 1} |0...01\rangle + ... + \psi_{\lambda n} |1...11\rangle$$

- Then by definition $U |\psi_\lambda\rangle = e^{i\lambda} |\psi_\lambda\rangle$

- Recall that $\left|det(e^{iX})\right| = \left|e^{tr(iX)}\right| = \left|e^{i}\right| = 1$
  - X is normalized w.r.t. its trace!
- Since U is a unitary matrix we can implement the corresponding quantum gate, as well as its controlled version
- Let $c-U$ be the controlled version of gate U
- Then
  $|0\rangle\,|\psi_\lambda\rangle \xrightarrow{c-U} |0\rangle\,|\psi_\lambda\rangle$
  $|1\rangle\,|\psi_\lambda\rangle \xrightarrow{c-U} |1\rangle\, e^{i\lambda}\,|\psi_\lambda\rangle$

- It can be proven that, from a mathematical point of view, quantum states $|1\rangle e^{i\lambda} |\psi_\lambda\rangle$ and $e^{i\lambda} |1\rangle |\psi_\lambda\rangle$ are **the same state**
- The phase factor is *kicked back* from the controlled register to the control register

- The state $e^{i\lambda}|1\rangle|\psi_\lambda\rangle$ now encodes λ in its phase, but as a *global* phase shift, which is not measurable
- In order to introduce a *relative* phase shift that can then be measured, we must simply put the control register in a superposition prior to the application of the $c-U$
    - *e.g.* we apply the Hadamard gate!

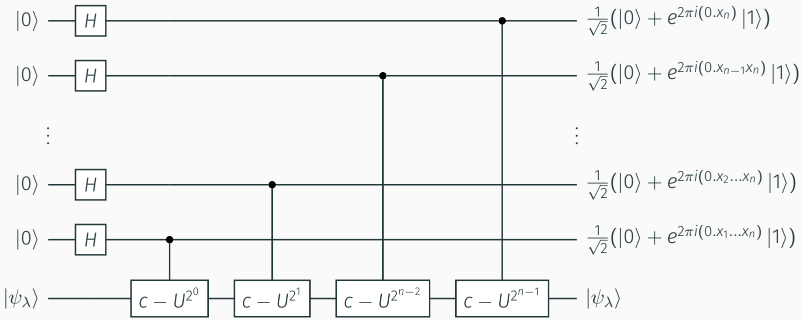
$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|\psi_\lambda\rangle \xrightarrow{c-U} \frac{1}{\sqrt{2}}(|0\rangle + e^{i\lambda}|1\rangle)|\psi_\lambda\rangle$$

- As a last step, consider $c-U$ raised to power k
- Eigenvalue of $c-U^k$ corresponding to $|\psi_\lambda\rangle$ will be $e^{ik\lambda}$
- Applying $c-U^k$ using different control qubits, with k equals to **increasing powers of 2**, we obtain our goal configuration

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{ik\lambda}|1\rangle)|\psi_\lambda\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.x_{(\log_2 k+1)}\dots x_n)}|1\rangle)|\psi_\lambda\rangle$$

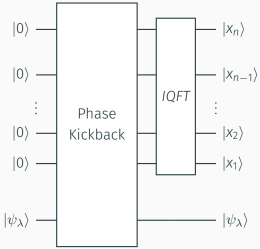
– Note that $|\psi_\lambda\rangle$ is left unchanged after every application, thus we can keep reusing it

Circuit for the Phase Kickback



QPE

Putting everything together



- This is how standard QPE works for computing an eigenvalue when the corresponding eigenvector is known
- In our case, however, **we don't know any of the eigenvectors of X a priori**, and therefore we cannot prepare the state $|\psi_\lambda\rangle$

Quantum PCA

- How can we exploit Quantum Phase Estimation to perform eigendecomposition in our scenario?

– **Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors** (Abrams & Lloyd, 1999) [1]

– **Quantum principal component analysis** (Lloyd, Mohseni & Rebentrost, 2014) [2]

  – **Towards Pricing Financial Derivatives with an IBM Quantum Computer** (Martin et al., 2019) [3]

Seth Lloyd

Quantum PCA

- As it happens, precise knowledge of $|\psi_\lambda\rangle$ is **not** required in order to estimate λ
- It suffices to have an estimate ψ_a, such that $|\langle\psi_a|\psi_\lambda\rangle|^2$ is not exponentially small (w.r.t. problem size), to measure the value of λ in a polynomial number of trials
- Even more interestingly, when the *n* qubits are measured to be in state λ, the register that initially stored $|\psi_a\rangle$ collapses to state $|\psi_\lambda\rangle$!
- How is this explained?

Quantum PCA

- The state we are using as input can be written as a **linear combination** of the true eigenvectors of X, labelled $|\Phi_k\rangle$

$$|\psi_a\rangle = \sum_k c_k |\Phi_k\rangle$$

- QPE on this state then yields measurements of each eigenvalue λ_k with probability $|c_k|^2$, while making the initial vector collapse to the corresponding true eigenvector $|\Phi_k\rangle$

Quantum PCA

- This is why using an estimate of a true eigenvector gives us the corresponding eigenvalue with sufficient reliability
- In this case, ”unexpected” measurements are not just meaningless noise, but are in fact occurrences of different eigenvalues
- Thus, for any random initial vector $|\psi\rangle$, the output configuration will in general reveal **all eigenvector-eigenvalue couples**

Quantum PCA

- In addition, if we are able to obtain an approximation for an eigenvector using a random initial state, we can sequentially improve the result by using such approximation as the new initial state
- This brings us to a crucial point for the algorithm: **extracting information from the output state**

- After applying QPE the eigenvectors are entangled with their eigenvalues and encoded in $log_2(|\psi_\lambda|)$ qubits
- Recall Amplitude Encoding:

$$|\psi_\lambda\rangle = \psi_{\lambda 0}\,|0...00\rangle + \psi_{\lambda 1}\,|0...01\rangle + ... + \psi_{\lambda n}\,|1...11\rangle$$

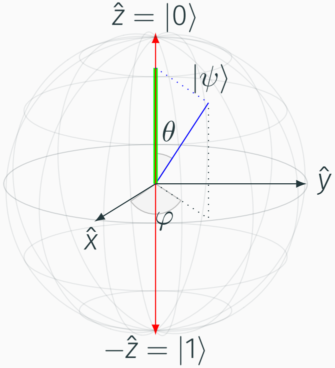
- Using data from N shots of the circuit, we can estimate these probabilities as relative frequencies $|\psi_{\lambda i}| = \sqrt{\frac{n_i}{N}}$
- How do we get the relative phases, *i.e.* detect signs?

Quantum Tomography

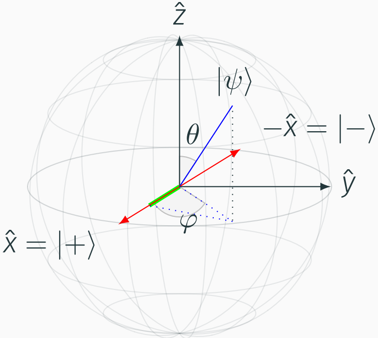
- The technique used to reconstruct information of a quantum state is called **Quantum State Tomography**
- Tomography is done by collecting and processing different types of measurements, or *observables*, of the system which correspond to various axes of the Bloch sphere

– Note that, still, the global phase of a state is not measurable, meaning that states $\alpha|0\rangle - \beta|1\rangle$ and $-\alpha|0\rangle + \beta|1\rangle$ are physically indistinguishable.
This is not relevant in our scenario as we are only interested in the direction of the vector and not in its sense

Quantum Tomography



Measurement on Z basis



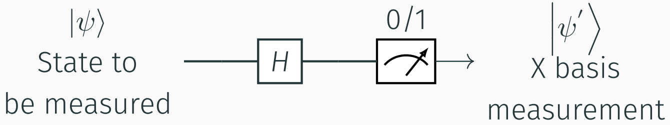
Measurement on X basis

Quantum Tomography

- Full Quantum State Tomography of a b-qubit system is in general computationally intensive
  - $O(4^b)$ different observables must be measured
  - Data processing involves common optimization algorithms
- However, we can exploit problem structure to do better
- In particular we assume to work with *real* amplitudes (not complex as in the general case)
  - Input state has real amplitudes
  - No rotations around the y axis performed by the algorithm in the ideal execution

Quantum Tomography

- We measure the register only in the Z basis (all qubits) and in the combinations of 1 qubit at a time in the X basis (*i.e.* $b+1$ settings)
- We obtain measurements in the X basis from ones in the *standard computational basis* (Z basis) by applying an Hadamard gate before measurement



- This operation maps the state $|+\rangle$ to output 0 and the state $|-\rangle$ to output 1

- Performing these measurements we can compare the probability of different outcomes in order to estimate the relative phases
- The total number of comparisons is $2^b - 1$, linear in the Hilbert space dimension
  - One for each amplitude!

Implementation and Analysis

Implementation

- We implement the algorithm in Python using Qiskit

```
def qpca(covmat, precision, initial, backend, req_shots)
```

- *covmat*: input covariance matrix
- *precision*: number of qubits to represent the eigenvalues
- *initial*: initial vector ($|\psi\rangle$)
- *backend*: to run the circuit (*qasm_simulator* by default)
- *req_shots*: shots of the circuit (8192 by default)

Demo

Here is a demo of the algorithm applied to the Iris dataset.

In order to analyze the algorithm and its complexity there are different points to take into account

- Implementation of gate e^{iX}
- Quantum Phase Estimation
- Quantum State Tomography
- Executions of the circuit

- Exponentiation of the covariance matrix is required to implement the (controlled) $U = e^{iX}$
- In Quantum Information, the process of finding a circuit that approximates such operator for a given Hamiltonian is called **Hamiltonian Simulation**
    - Note: also Hamiltonians are Hermitian matrices
- This was one of the original motivating problems of Quantum Computing proposed by Richard Feynman as a key part of simulating quantum mechanical systems.

Approaches to Simulation of non-sparse d-dimensional Hamiltonians

- Higher order Suzuki-Trotter expansion: $O(d \log d)$
- Lloyd-Mohseni-Rebentrost method: $O(\log d)$
  Assuming many copies of the Hamiltonian are available as quantum states (encoded in density matrix)
- Wang and Wossnig: $\tilde{O}(\sqrt{d})$
  Using a more practical qRAM data structure
  arXiv:1803.08273

Gate-complexity of Quantum Phase Estimation is impacted by the precision of the eigenvalue estimation (m qubits)

- IQFT is $O(m^2)$
- m calls to powers of $c - U$
- If $c - U^k = k$ times $c - U$, gate-complexity is 2^m
- Then overall gate-complexity: $O(m^2 + 2^m)$
- Note: complexity scales linearly with accuracy $M = 2^m$

Already discussed.

Our implementation for d dimensional eigenvectors encoded in $b = \lceil \log_2 d \rceil$ qubits

- $b + 1$ different observables to be measured
- $O(2^b) = O(d)$ comparisons on the results of such measurements

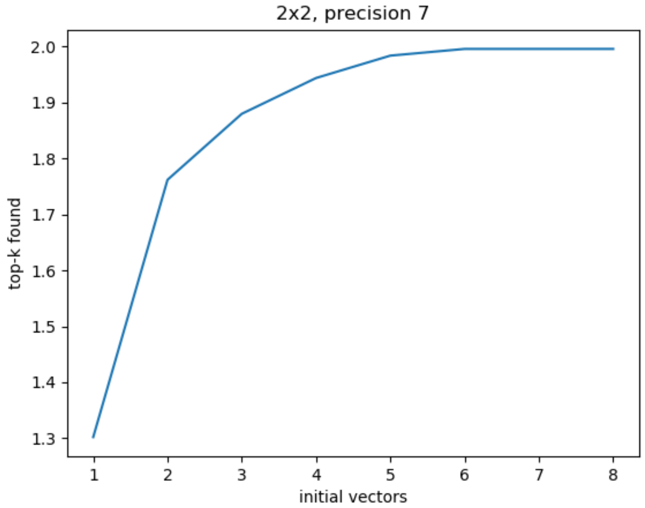
We know *what* we have to measure, but *how many* measurements should be taken?
Considering each eigenvalue/eigenvector couple

- Number of measurements in the *"all-Z basis"* scales only with precision of amplitude estimation
- Measurements in the *"one-X basis"* give clear results for strongly positive/negative amplitudes, ambiguous for values near 0

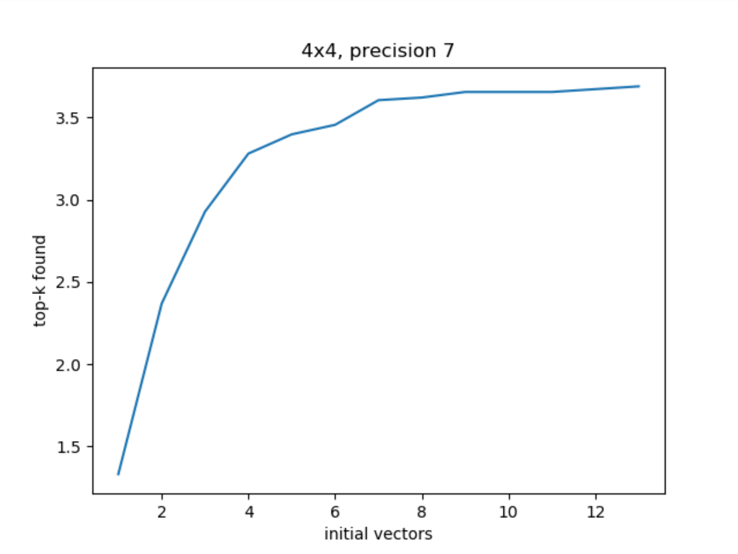
Recall: in principle, any random initialization reveals all eigenvalue/eigenvector couples.
In practice, some eigenvectors may have *exponentially small* contribution to an initial vector and appear rarely in output.
Different approaches

- Multiple initializations (random or by some criteria)
- Obtain rough estimate of eigenvector and iterate

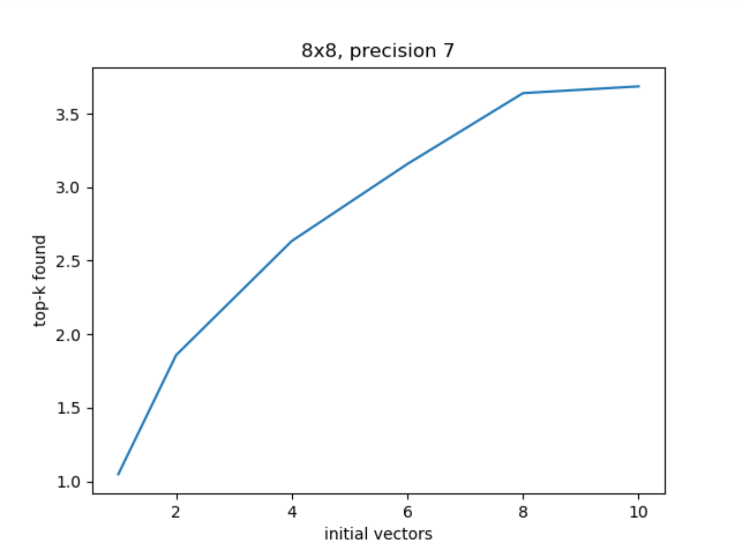
Some results for different numbers of random initializations



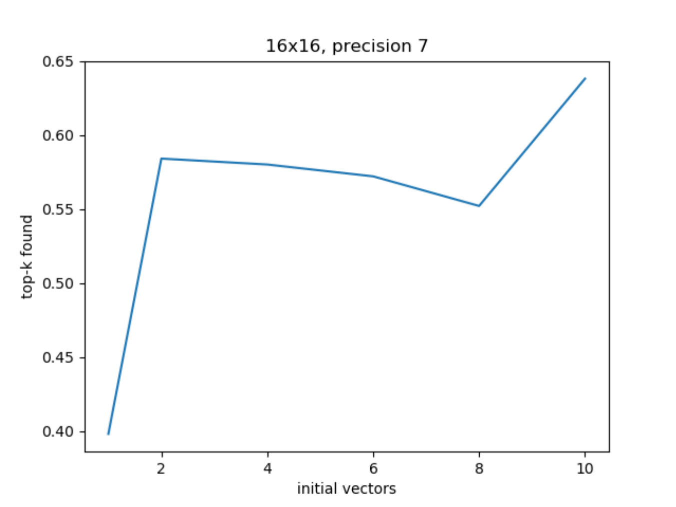
Analysis - *Results*



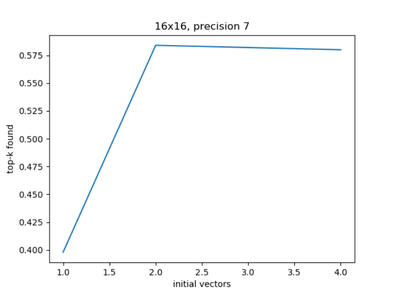
Analysis - *Results*

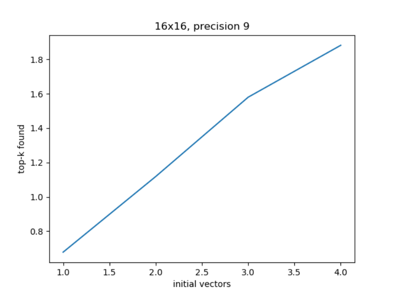


With 7 qubits precision, results on 16x16 covariance matrices start degrading.



Raising precision to 9 qubits makes results immediately improve





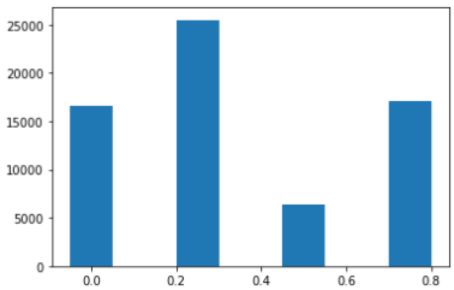
We test the algorithm on IBM's 5-qubit quantum device Vigo

- 2*x*2 covariance matrix
- 2 qubits precision for the eigenvalues

Real eigendecomposition

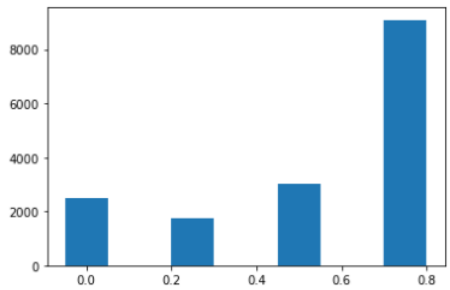
$$\lambda_1 = 0.84212698,\ v_1 = \begin{bmatrix} 0.91854462 & -0.39531732 \end{bmatrix}$$
$$\lambda_2 = 0.15787302,\ v_2 = \begin{bmatrix} 0.39531732 & 0.91854462 \end{bmatrix}$$

Histogram of eigenvalues found using 4 random initial vectors



Vector corresponding to 0.75, $v_a = \begin{bmatrix} 0.85450074 & -0.51945017 \end{bmatrix}$

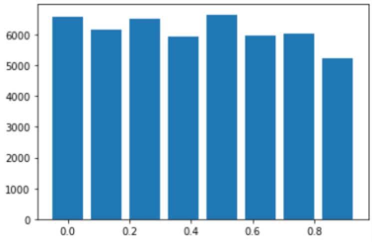
Running the algorithm using v_a as initial vector now measures



and improved vector $v = \begin{bmatrix} 0.91533142 & -0.40270137 \end{bmatrix} \approx v_1$

Analysis - *Results on Real Device*

- Iterative Approach proves effective with today's noisy quantum computers
- For 2x2 matrix we obtained decent results
- Attempts on 4x4 with 3 qubit precision are already too noisy



References

- Daniel S. Abrams, Seth Lloyd *A quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors.* `https://arxiv.org/pdf/quant-ph/9807070v1.pdf`
- Seth Lloyd , Masoud Mohseni and Patrick Rebentrost *Quantum principal component analysis.* `https://arxiv.org/pdf/1307.0401.pdf`
- Ana Martin, Bruno Candelas, Ángel Rodríguez-Rozas et al. *Towards Pricing Financial Derivatives with an IBM Quantum Computer.* `https://arxiv.org/pdf/1904.05803.pdf`