HHL Algorithm

Linear system of equations

Given an invertible n-by-n matrix A and an n dimensional vector b find an x such that

$$Ax = b$$

Applications:

- Least squares fitting
- Numerical methods for solving differential equations
- Solving optimization problems (linear programming)

Solving a linear system

$$Ax = b$$

Methods:

- Gaussian elimination $O(n^3)$.

- Fast matrix multiplication $O(n^\omega)$
where $\omega < 2.372864$.

- Iterative methods

  Conjugate Gradient $A \succ 0$

  $$O(\mathrm{nnz}(A)\sqrt{\kappa(A)}\log(1/\varepsilon))$$

Quantum Version

Given:

- Oracle access to an invertible matrix A .

- A unitary that prepares quantum state $|b\rangle$ encoding a vector b in its amplitudes (assume $\|b\| = 1$).

Output: A quantum state $|\tilde{x}\rangle$ approximately encoding a vector x such that $Ax = b$.

Specifically, we want $\||x\rangle - |\tilde{x}\rangle\| \leq \varepsilon$.

Harrow, Hassidim, and Lloyd give a quantum algorithm for this problem arXiv:0811.3171.

Differences

It is important to note that the output is not the same in the classical and quantum versions.

The classical output is the vector x.

In the quantum version we only get a state $|x\rangle$.

We can then sample from $|x\rangle$ or estimate statistics of it by making measurements.

Sparse Matrix

- Oracle access to a matrix A .

We assume that A is s -sparse: each row has at most s nonzero entries.

$$O_A : |i,j\rangle|0\rangle \mapsto |i,j\rangle|A(i,j)\rangle$$

$$O_{A,loc} : |j\rangle|\ell\rangle \mapsto |j\rangle|\nu(j,\ell)\rangle$$

Where the ℓ^{th} nonzero entry in row j is in column $\nu(j,\ell)$.

Preparation of $|b\rangle$

- We will need to generate copies of $|b\rangle$ in the algorithm so we assume we have a unitary that does this.

For arguing about the complexity we suppose this unitary can be constructed with T_B gates.

Condition Number

- We assume that A is not just invertible, but far away from being singular.

Let's assume that A is Hermitian. Let $\lambda_1 \geq \cdots \geq \lambda_n$ be its eigenvalues.

$$\sigma_1 = \max_{\lambda \in \{\lambda_1, \ldots, \lambda_n\}} |\lambda|$$

$$\sigma_n = \min_{\lambda \in \{\lambda_1, \ldots, \lambda_n\}} |\lambda|$$

For non-singular A, the condition number is

$$\kappa(A) = \frac{\sigma_1}{\sigma_n}$$

Eigenvalues of inverse

Let's assume that A is Hermitian. Let $\lambda_1 \geq \cdots \geq \lambda_n$ be its eigenvalues.

Question: What are the eigenvalues and eigenvectors of A^{-1}?

Condition Number

Recall the spectral norm of a matrix:

$$\|A\| = \sup_x \frac{\|Ax\|}{\|x\|}$$

Another way to state condition number is

$$\kappa = \|A\| \cdot \|A^{-1}\|$$

Say you make a small (relative) change in x, by how much can Ax change, relatively?

This is measured by the condition number.

$$\|A\| = \sup_x \frac{\|Ax\|}{\|x\|}$$

Say you make a small (relative) change in x, by how much can Ax change, relatively?

$$\sup_{\delta x} \frac{\|A(x+\delta x) - Ax\|}{\|Ax\|} \Big/ \frac{\|\delta x\|}{\|x\|}$$

$$= \sup_{\delta x} \frac{\|A\delta x\|}{\|\delta x\|} \frac{\|x\|}{\|Ax\|}$$

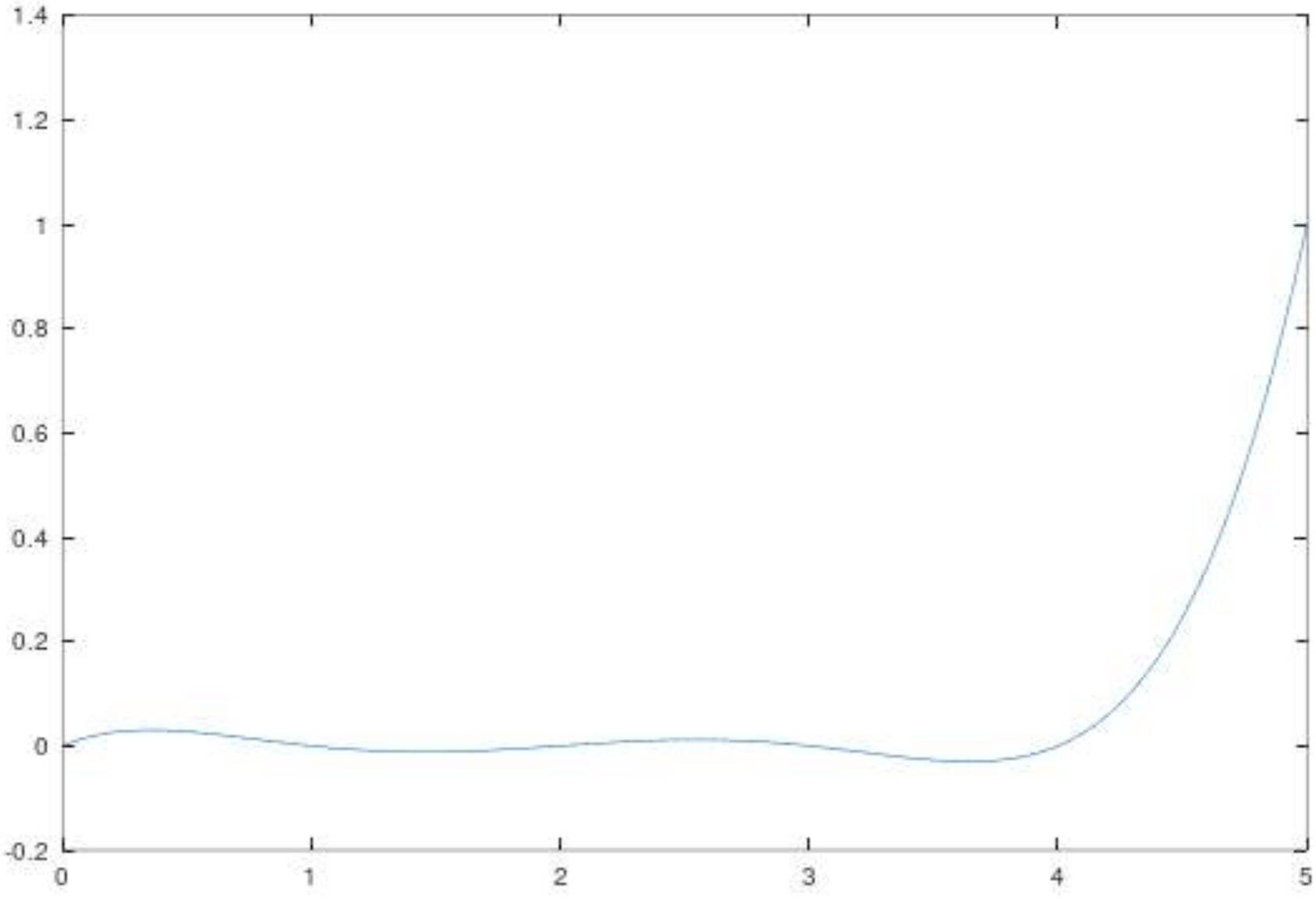
$$\leq \|A\| \cdot \|A^{-1}\|$$

A high condition number means low accuracy in floating point computations.

Example: Polynomial Interpolation

Say we want to find the coefficients of a degree 5 univariate polynomial $p(x)$ such that

$$p(0) = 0$$
$$p(1) = 0$$
$$p(2) = 0$$
$$p(3) = 0$$
$$p(4) = 0$$
$$p(5) = 1$$



Example: Polynomial Interpolation

The polynomial is $\alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 + \alpha_4 x^4 + \alpha_5 x^5$ where α is the solution to the linear system:

$$
\begin{aligned}
p(0) &= 0 \\
p(1) &= 0 \\
p(2) &= 0 \\
p(3) &= 0 \\
p(4) &= 0 \\
p(5) &= 1
\end{aligned}
\qquad
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 \\
2 & 4 & 8 & 16 & 32 \\
3 & 9 & 27 & 81 & 243 \\
4 & 16 & 64 & 256 & 1024 \\
5 & 25 & 125 & 625 & 3125
\end{bmatrix}
\begin{bmatrix}
\alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5
\end{bmatrix}
=
\begin{bmatrix}
0 \\ 0 \\ 0 \\ 0 \\ 1
\end{bmatrix}
$$

The condition number of this matrix is approximately 55,674.

Plan

- How the HHL algorithm works.

- Applications? How to deal with the assumptions.

- Quantum-inspired classical algorithms: a fair comparison with classical algorithms.

Ingredients

First let's review the tools that will go into the HHL algorithm.

- Hamiltonian simulation of sparse matrices.
- Phase estimation.
- Amplitude amplification.

Hamiltonian Simulation

If A is a 2^n-by-2^n Hermitian s-sparse matrix then we can implement $e^{-\mathrm{i}At}$ up to error ε in the spectral norm with

arXiv:1606.02685

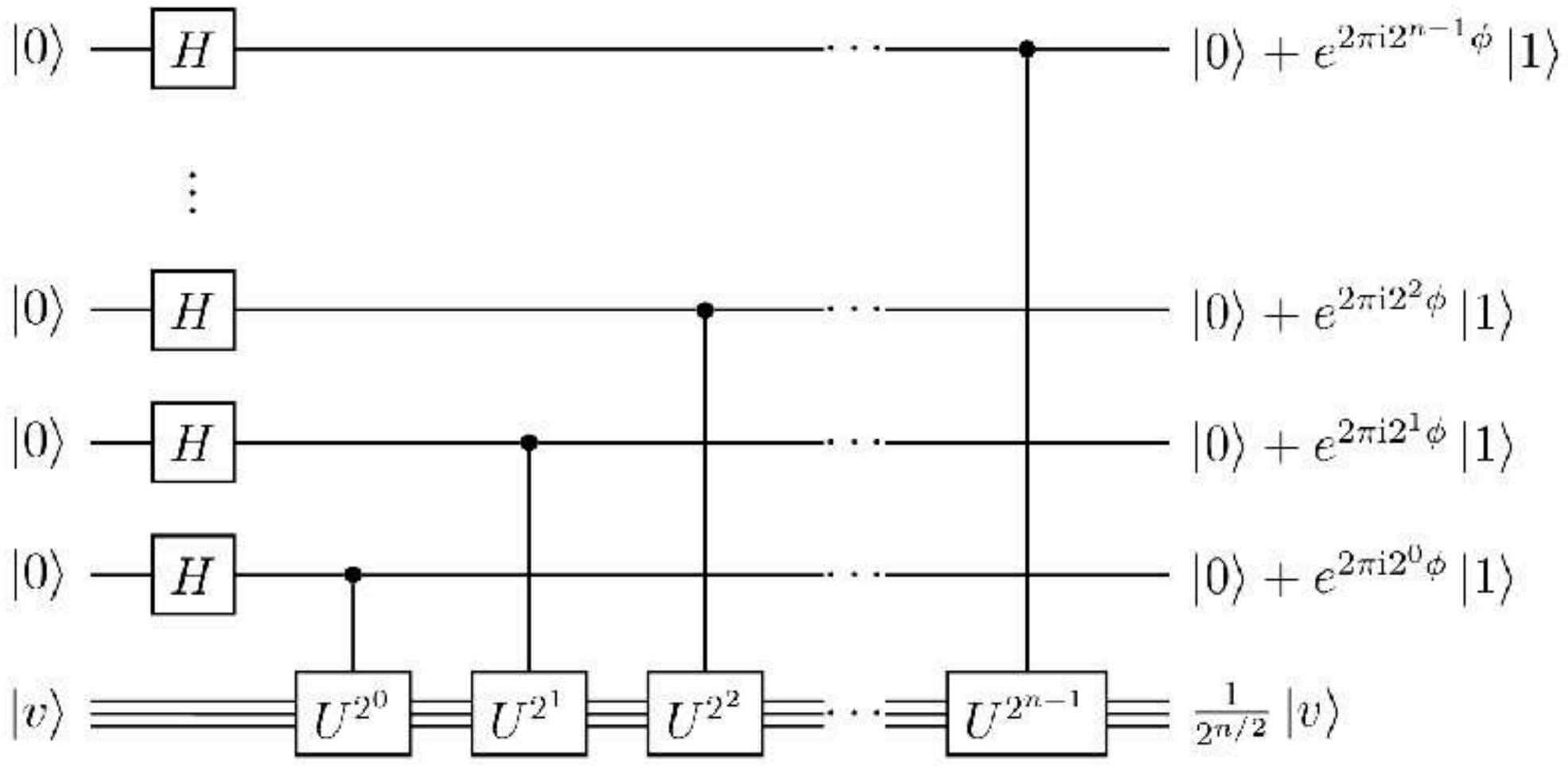
$$O\left(st\|A\|_\infty + \frac{\log(1/\varepsilon)}{\log\log(1/\varepsilon)}\right)$$

queries to A and total number of gates larger by a factor of n.

This can be done via the quantum signal processing approach discussed at the end of Maria's lecture.

Phase Estimation

$$U|v\rangle = e^{2\pi i\phi}|v\rangle$$



To estimate ϕ to error $1/2^n$ we need $\Theta(2^n)$ applications of U .

Amplitude Amplification

We have not explicitly talked about amplitude amplification yet.

Suppose a quantum algorithm $\mathcal{A}$ is such that

$$\mathcal{A}|0^m\rangle = \alpha_0|0\rangle|\psi_0\rangle + \alpha_1|1\rangle|\psi_1\rangle$$

and a lower bound $p \leq |\alpha_0|^2$ is known.

There is a quantum algorithm that makes $O(1/\sqrt{p})$ calls to $\mathcal{A}$ and $\mathcal{A}^{-1}$ and outputs $|0\rangle|\psi_0\rangle$ with prob. at least $9/10$.

High-Level Overview

We will assume that $\|A\| = 1$, thus all eigenvalues have magnitude at least $1/\kappa$.

Let $v_1, \ldots, v_n$ be an orthonormal set of eigenvectors for A, where v_j corresponds to eigenvalue λ_j.

We can write

$$A = \sum_{j=1}^{n} \lambda_j v_j v_j^* \qquad A^{-1} = \sum_{j=1}^{n} \frac{1}{\lambda_j} v_j v_j^*$$

We can also express the RHS b in the basis of eigenvectors.

$$b = \sum_{j=1}^{n} \beta_j v_j$$

The algorithm does not need to know these coefs, we just work in the eigenvector basis for the analysis.

We want to obtain (up to normalization) the state

$$\sum_{i=1}^{n} \frac{\beta_j}{\lambda_j} |v_j\rangle$$

We can't just apply A^{-1} to $|b\rangle$ as it might not be unitary.

However,

$$e^{\mathrm{i}A} = \sum_j e^{\mathrm{i}\lambda_j} v_j v_j^*$$

is unitary and has the same eigenvectors as A and A^{-1}.

We can implement $U = e^{\mathrm{i}A}$ and its powers $U = e^{\mathrm{i}At}$ using Hamiltonian simulation.

This lets us do phase estimation on U.

High-Level Overview

Action of the algorithm on an eigenvector $|v_j\rangle$.

1) Start in the state $|v_j\rangle|0\rangle|0\rangle$.

2) Do phase estimation on $U = e^{\mathrm{i}A}$ **with** $|v_j\rangle$

Suppose this works perfectly

$$|v_j\rangle|0\rangle|0\rangle \rightarrow |v_j\rangle|\lambda_j\rangle|0\rangle$$

High-Level Overview

2) Do phase estimation on $U = e^{\mathrm{i}A}$ with $|v_j\rangle$

Suppose this works perfectly

$$|v_j\rangle|0\rangle|0\rangle \rightarrow |v_j\rangle|\lambda_j\rangle|0\rangle$$

Really want an estimate θ_j with $|\theta_j - \lambda_j| \leq \frac{\varepsilon}{\kappa}$.

The query cost of this step will be roughly $O(s\kappa/\varepsilon)$.

High-Level Overview

3) Do a rotation on 3rd register conditioned on 2nd.

$$|v_j\rangle|\lambda_j\rangle|0\rangle \rightarrow |v_j\rangle|\lambda_j\rangle \left(\frac{1}{\kappa\lambda_j}|0\rangle + \sqrt{1 - \frac{1}{\kappa^2\lambda_j^2}}|1\rangle \right)$$

We know that $\kappa|\lambda_j| \geq 1$ so this is a valid quantum state.

4) Undo the phase estimation and drop 2nd register.

$$\frac{1}{\kappa\lambda_j}|v_j\rangle|0\rangle + \sqrt{1 - \frac{1}{\kappa^2\lambda_j^2}}|v_j\rangle|1\rangle$$

High-Level Overview

When we do these 4 steps on $|b\rangle = \sum_{j=1}^{n} \beta_j |v_j\rangle$ we end up with

$$\frac{1}{\kappa} \left(\sum_{j=1}^{n} \frac{\beta_j}{\lambda_j} |v_j\rangle \right) |0\rangle + |\phi\rangle|1\rangle$$

We know that $\sum_{j=1}^{n} \frac{|\beta_j|^2}{\lambda_j^2} \geq \sum_{j=1}^{n} |\beta_j|^2 = 1$ because $\|A\| = 1$.

The norm of part of state ending in $|0\rangle$ is at least $1/\kappa$.

High-Level Overview

$$\frac{1}{\kappa}\left(\sum_{j=1}^{n}\frac{\beta_j}{\lambda_j}|v_j\rangle\right)|0\rangle+|\phi\rangle|1\rangle$$

After $O(\kappa)$ rounds of amplitude amplification, measuring the last qubit gives a state proportional to

$$\left(\sum_{j=1}^{n}\frac{\beta_j}{\lambda_j}|v_j\rangle\right)|0\rangle$$

with constant probability.

Complexity

The 4 steps can be done with roughly $O(\kappa s/\varepsilon)$ many queries to A and $O(\log(n)\kappa s/\varepsilon + T_B)$ other gates.

The total cost with amplitude amplification is

$O(\kappa^2 s/\varepsilon)$ queries

$O(\log(n)\kappa^2 s/\varepsilon + \kappa T_B)$ other gates

The $1/\varepsilon$ factor has since been improved to $\log(\kappa/\varepsilon)$. "Variable time amplitude amplification" can reduce κ dependence to linear.

arXiv:1511.02306

arXiv:1010.4458

Complexity

$O(\kappa^2 s/\varepsilon)$ queries

$O(\log(n)\kappa^2 s/\varepsilon + \kappa T_B)$ other gates

The important thing is the dependence on n only goes like $\log(n)$.

If all other parameters are polylog, can get polylog complexity overall.

Notes

- We can replace sparsity of A with any condition that implies efficient Hamiltonian simulation.

- For a polylog conditioned and $O(1)$-sparse matrix A, estimating if the first entry of $A^{-1}b/\|A^{-1}b\|$ is $\geq 2/3$ or $\leq 1/3$ is BQP complete.

We do not expect this problem to be efficiently solvable classically in general.

Perspective

The HHL algorithm has been enormously influential especially for the area of quantum machine learning.

There is still a lot of active discussion on the applications of this algorithm.

Because of the caveats we now discuss further, in my opinion there still have not been killer applications.

Perspective

• Aaronson, "Quantum Machine Learning Algorithms: Read the fine print"

• Biamonte et al., "Quantum Machine Learning" arXiv:1611.09347

• Ewin Tang talk at the Simons Institute "On quantum linear algebra for machine learning" https://www.youtube.com/watch?v=OE5a0Mgcgwc

And the panel discussion following this talk.

Finding Applications

There are 3 primary difficulties in finding applications of HHL.

1) State preparation: there needs to be an efficient way to prepare copies of $|b\rangle$.

If you have to write/read b classically, exponential speedup is gone.

2) Solution readout:

What can we do with $|x\rangle$?

Say $p = |\langle x|M|x\rangle|^2$ how to estimate this?

Can take $\frac{1}{\varepsilon^2 p}$ trials to estimate within εp.

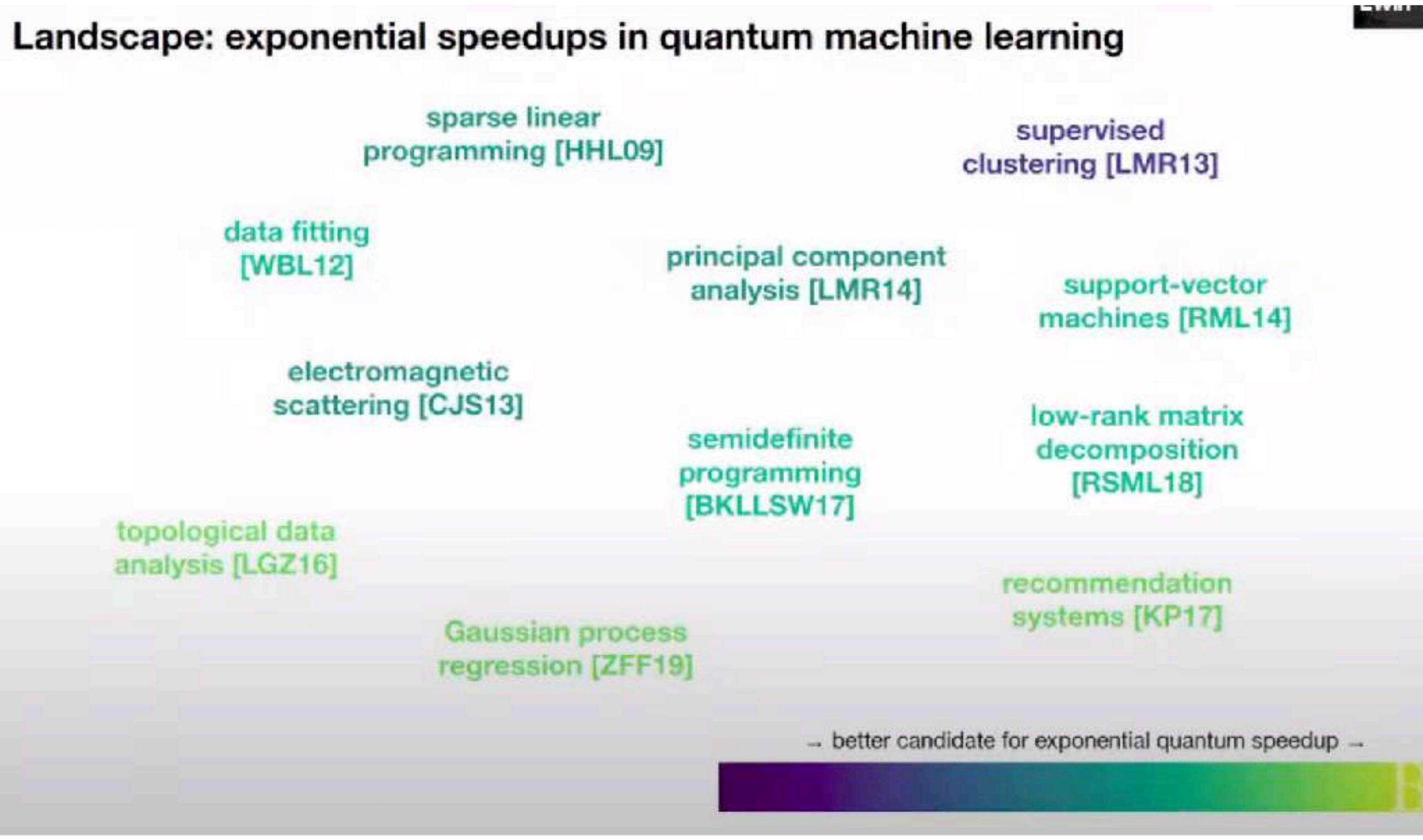
Clader et al. show how to estimate $|\langle z|x\rangle|$ using amplitude estimation (quadratic speedup).

Need to find application where this suffices and can be done efficiently.

3) Condition number:

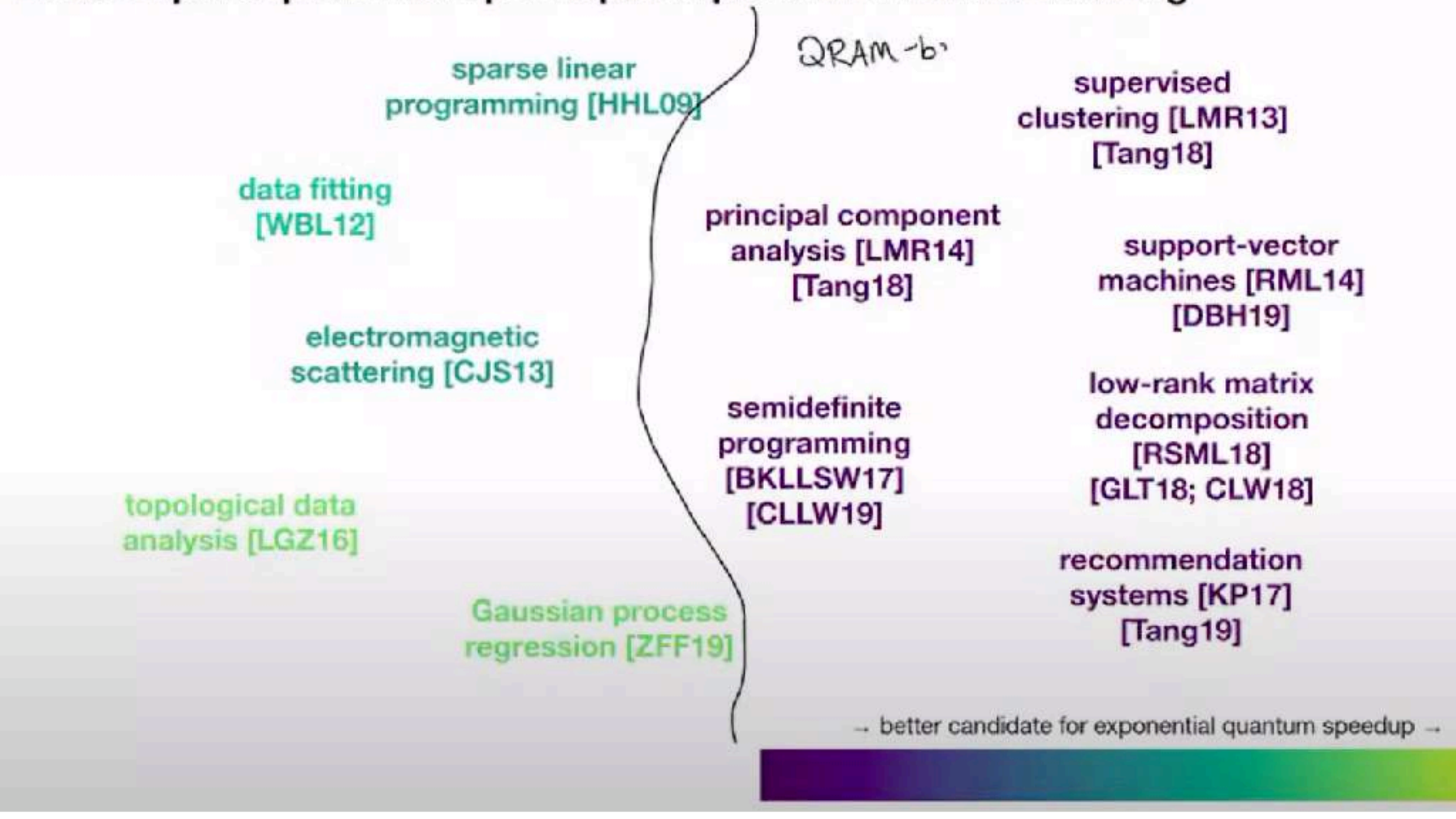
To be polylog overall, need a polylog condition number.

This can be difficult to find/prove in interesting applications.



Slide from Ewin Tang's talk

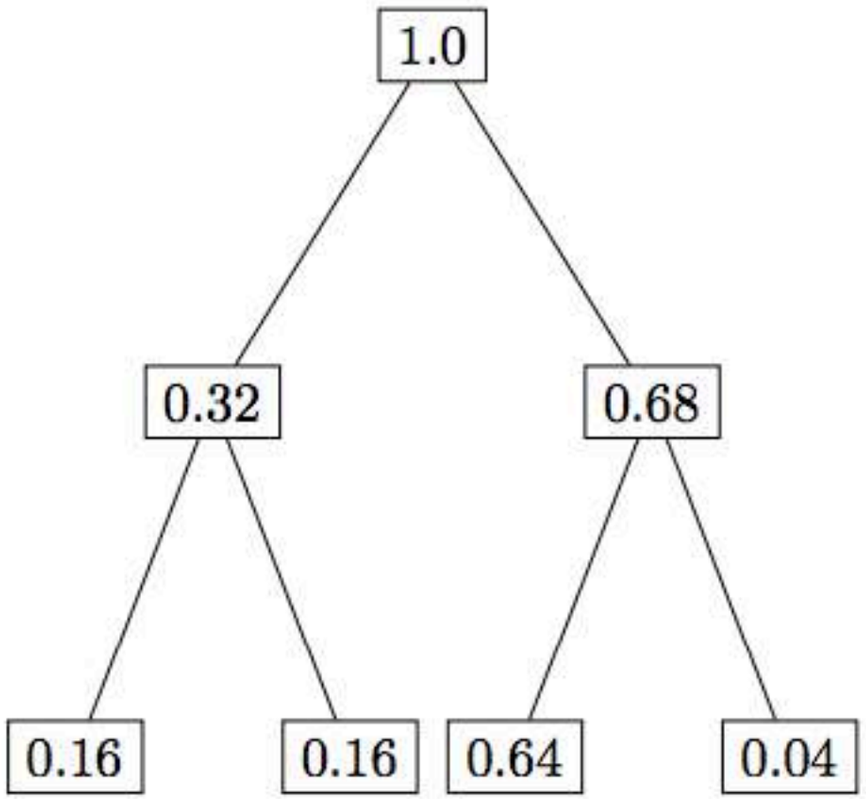
https://www.youtube.com/watchv=OE5a0Mgcgwc



Result of quantum inspired classical algorithms

How to prepare $|b\rangle$

QRAM: There is a classical data structure that through querying in superposition we can efficiently prepare $|b\rangle$.



Let $|\phi\rangle = 0.4\,|00\rangle + 0.4\,|01\rangle + 0.8\,|10\rangle + 0.2|11\rangle$.

- Rotation on qubit 1:
  $|0\rangle\,|0\rangle \to (\sqrt{0.32}\,|0\rangle + \sqrt{0.68}\,|1\rangle)\,|0\rangle$
- Rotation on qubit 2 conditioned on qubit 1:

$$(\sqrt{0.32}\,|0\rangle + \sqrt{0.68}\,|1\rangle)\,|0\rangle \to$$
$$\sqrt{0.32}\,|0\rangle\,\frac{1}{\sqrt{0.32}}(0.4\,|0\rangle + 0.4\,|1\rangle) +$$
$$\sqrt{0.68}\,|1\rangle\,\frac{1}{\sqrt{0.68}}(0.8\,|0\rangle + 0.2\,|1\rangle)$$

Figure 1: Vector state preparation illustrated for 4-dimensional state $|\phi\rangle$.

From Kerenidis and Prakash: arXiv:1603.08675.

Fair Comparison

QRAM: There is a classical data structure that through querying in superposition we can efficiently prepare $|b\rangle$.

Many QML algorithms assume access to such a data structure.

Ewin Tang's idea: For fair comparison, classical algorithms should be allowed to use this data structure too.

This is a powerful classical primitive.

Sampling + Query Access

Let $v \in \mathbb{C}^n$. Sampling + query (SQ) access to v allows

1) Obtain independent samples $j \in [n]$ with prob. $|v_j|^2 / \|v\|^2$.

2) Query entries of v.

3) Query $\|v\|$.

SQ access to a matrix A allows SQ access to each row, and to the vector of row norms.

Quantum-Inspired

Let $\kappa_F = \|A\|_F \cdot \|A^{-1}\|$.

Thm: Suppose we have SQ access to b and an s-sparse matrix A. There is a classical randomized algorithm that returns $\tilde{x}$ with

$$\|\tilde{x} - A^{-1}b\| \leq \varepsilon \|A^{-1}b\|$$

in time $O(s\kappa_F^2 \log(1/\varepsilon))$.

arXiv:2103.10309