Eigenvalue decomposition on a quantum computer

Changpeng Shao

School of Mathematics, University of Bristol, UK
joint work with Jin-Peng Liu (UC Berkeley)
arXiv:2010.15027

Computational Mathematics for Quantum Technologies
1-5, August 2022, Bath, UK

Quantum computing

Quantum computers are designed to solve some problems faster than classical computers.

Quantum computing

Quantum computers are designed to solve some problems faster than classical computers.

Quantum computers are particularly good at linear algebra problems, e.g.,

1. Solving linear systems
2. Functions of matrices
3. Singular value decomposition
4. Eigenvalue decomposition

Quantum computing

Quantum computers are designed to solve some problems faster than classical computers.

Quantum computers are particularly good at linear algebra problems, e.g.,

1. Solving linear systems
2. Functions of matrices
3. Singular value decomposition
4. Eigenvalue decomposition

This leads to wide applications in many areas, e.g.,

1. Solving differential equations
2. Solving polynomial equations
3. Machine learning

This talk

In this talk, I will focus on the problem of eigenvalue decomposition.

This talk

In this talk, I will focus on the problem of eigenvalue decomposition.

Main motivation.
With quantum SVD, we can solve many fundamental problems in linear algebra, e.g., solving linear systems (matrix multiplication), but usually not eigenvalues problems. For eigenvalue problems, we currently know how to solve them for normal matrices (e.g., Hermitian, unitary) on a quantum computer.

This talk

In this talk, I will focus on the problem of eigenvalue decomposition.

Main motivation.
With quantum SVD, we can solve many fundamental problems in linear algebra, e.g., solving linear systems (matrix multiplication), but usually not eigenvalues problems. For eigenvalue problems, we currently know how to solve them for normal matrices (e.g., Hermitian, unitary) on a quantum computer.

Goal.
Explore the potential of quantum computers on eigenvalue problems.

Quantum eigenvalue decomposition

Let H be an $n \times n$ Hermitian matrix. Suppose the eigenpairs are $(\lambda_j, |u_j\rangle), j \in [n]$.

Quantum eigenvalue decomposition

Let H be an $n \times n$ Hermitian matrix. Suppose the eigenpairs are $(\lambda_j, |u_j\rangle), j \in [n]$.

Classically, one goal is to compute all the eigenpairs, i.e., compute the decomposition $H = UDU^{-1}$. But quantumly,

Quantum eigenvalue decomposition

Let H be an $n \times n$ Hermitian matrix. Suppose the eigenpairs are $(\lambda_j, |u_j\rangle), j \in [n]$.

Classically, one goal is to compute all the eigenpairs, i.e., compute the decomposition $H = UDU^{-1}$. But quantumly,

Goal: achieve the map

$$\sum_{j=1}^{n} \beta_j |u_j\rangle \to \sum_{j=1}^{n} \beta_j |\tilde{\lambda}_j\rangle |u_j\rangle,$$

where $|\tilde{\lambda}_j - \lambda_j| \leq \varepsilon$ for all j.

This is achieved by quantum phase estimation.
[Kitaev, arXiv:quant-ph/9511026, 1995]

Quantum phase estimation

Set $D = O(1/\varepsilon)$ and suppose the input is $|u_k\rangle$ for simplicity.

Quantum phase estimation

Set $D = O(1/\varepsilon)$ and suppose the input is $|u_k\rangle$ for simplicity.

1. Prepare

$$\frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} |j\rangle \otimes |u_k\rangle.$$

Quantum phase estimation

Set $D = O(1/\varepsilon)$ and suppose the input is $|u_k\rangle$ for simplicity.

1. Prepare
$$\frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} |j\rangle \otimes |u_k\rangle.$$

2. Apply control e^{iHj}
$$\frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} |j\rangle \otimes e^{iHj}|u_k\rangle = \frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} e^{i\lambda_k j}|j\rangle \otimes |u_k\rangle.$$

Quantum phase estimation

Set $D = O(1/\varepsilon)$ and suppose the input is $|u_k\rangle$ for simplicity.

1. Prepare
$$\frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} |j\rangle \otimes |u_k\rangle.$$

2. Apply control e^{iHj}
$$\frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} |j\rangle \otimes e^{iHj}|u_k\rangle = \frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} e^{i\lambda_k j}|j\rangle \otimes |u_k\rangle.$$

3. Apply quantum Fourier transform to $|j\rangle$
$$\sum_{l=0}^{D-1} \left(\frac{1}{D} \sum_{j=0}^{D-1} e^{i(\lambda_k - \frac{2\pi l}{D})j} \right) |l\rangle \otimes |u_k\rangle \approx |\tilde{\lambda}_k\rangle |u_k\rangle.$$

Quantum phase estimation

The complexity depends on the complexity $O(T)$ of implementing of e^{iHj}, which is known as Hamiltonian simulation. $\rightarrow$ $O(T/\varepsilon)$

Quantum phase estimation

The complexity depends on the complexity $O(T)$ of implementing of e^{iHj}, which is known as Hamiltonian simulation. $\rightarrow$ $O(T/\varepsilon)$

It is efficient ($T = \text{polylog}(n)$) when we can efficiently encode the matrix into the quantum circuit. $\rightarrow$ Block-encoding

$$U = \begin{pmatrix} H & \cdot \\ \cdot & \cdot \end{pmatrix},$$

where U is unitary. This is the case when H is sparse.

Quantum phase estimation

The complexity depends on the complexity $O(T)$ of implementing of e^{iHj}, which is known as Hamiltonian simulation. $\rightarrow$ $O(T/\varepsilon)$

It is efficient ($T = \text{polylog}(n)$) when we can efficiently encode the matrix into the quantum circuit. $\rightarrow$ Block-encoding

$$U = \begin{pmatrix} H & \cdot \\ \cdot & \cdot \end{pmatrix},$$

where U is unitary. This is the case when H is sparse.

The algorithm can be generalised to normal matrices.
[Low, Chuang, arXiv:1610.06546], [S, arXiv:1912.08015].

Quantum phase estimation

The complexity depends on the complexity $O(T)$ of implementing of e^{iHj}, which is known as Hamiltonian simulation. $\rightarrow$ $O(T/\varepsilon)$

It is efficient ($T = \text{polylog}(n)$) when we can efficiently encode the matrix into the quantum circuit. $\rightarrow$ Block-encoding

$$U = \begin{pmatrix} H & \cdot \\ \cdot & \cdot \end{pmatrix},$$

where U is unitary. This is the case when H is sparse.

The algorithm can be generalised to normal matrices.
[Low, Chuang, arXiv:1610.06546], [S, arXiv:1912.08015].

Question: What if H is not normal? How about generalised eigenvalue problems?

Generalised eigenvalue problems

Given two matrices A, B, we aim to compute all pairs $(\lambda, |u\rangle)$ such that

$$A|u\rangle = \lambda B|u\rangle.$$

Generalised eigenvalue problems

Given two matrices A, B, we aim to compute all pairs $(\lambda, |u\rangle)$ such that

$$A|u\rangle = \lambda B|u\rangle.$$

Goal: generalise quantum phase estimation, i.e., achieve the map for GEP

$$\sum_{j=1}^{n} \beta_j |u_j\rangle \to \sum_{j=1}^{n} \beta_j |\tilde{\lambda}_j\rangle |u_j\rangle,$$

where $|\tilde{\lambda}_j - \lambda_j| \leq \varepsilon$ for all j.

Generalised eigenvalue problems

A particularly important case: A is symmetric and B is positive semi-definite.

A standard algorithm:

$$(B^{-1/2}AB^{-1/2})B^{1/2}|u\rangle = \lambda B^{1/2}|u\rangle.$$

The matrix $B^{-1/2}AB^{-1/2}$ is symmetric. $\rightarrow$ QPE (some advanced quantum linear algebra techniques are required).

Can we do more?

[Parker, Joseph, arXiv:2002.08497].

Generalised eigenvalue problems

GEP has some "unfriendly" properties for a quantum computer:

- It is possible that for all λ we have $A|u\rangle = \lambda B|u\rangle$, e.g., $A|u\rangle = 0, B|u\rangle = 0$.
- When B is singular, then ∞ can be an eigenvalue, e.g., $A|u\rangle \neq 0, B|u\rangle = 0$.

Generalised eigenvalue problems

GEP has some "unfriendly" properties for a quantum computer:

- It is possible that for all λ we have $A|u\rangle = \lambda B|u\rangle$, e.g., $A|u\rangle = 0, B|u\rangle = 0$.
- When B is singular, then ∞ can be an eigenvalue, e.g., $A|u\rangle \neq 0, B|u\rangle = 0$.

Assumptions:

- B is invertible

  $\rightarrow$ The eigenpairs are finite.

Generalised eigenvalue problems

GEP has some "unfriendly" properties for a quantum computer:

- It is possible that for all λ we have $A|u\rangle = \lambda B|u\rangle$, e.g., $A|u\rangle = 0, B|u\rangle = 0$.
- When B is singular, then ∞ can be an eigenvalue, e.g., $A|u\rangle \neq 0, B|u\rangle = 0$.

Assumptions:

- B is invertible

  $\rightarrow$ The eigenpairs are finite.

- $B^{-1}A$ is diagonalizable (not necessary)

  $\rightarrow$ The eigenvectors are linearly independent so that any vector is a unique linear combination of them.

Generalised eigenvalue problems

GEP has some "unfriendly" properties for a quantum computer:

- It is possible that for all λ we have $A|u\rangle = \lambda B|u\rangle$, e.g., $A|u\rangle = 0, B|u\rangle = 0$.
- When B is singular, then ∞ can be an eigenvalue, e.g., $A|u\rangle \neq 0, B|u\rangle = 0$.

Assumptions:

- B is invertible

  $\rightarrow$ The eigenpairs are finite.

- $B^{-1}A$ is diagonalizable (not necessary)

  $\rightarrow$ The eigenvectors are linearly independent so that any vector is a unique linear combination of them.

- all eigenvalues are real

  $\rightarrow$ The focus of this talk.

Basic idea of generalising QPE

Use a similar idea to quantum phase estimation.

Basic idea of generalising QPE

Use a similar idea to quantum phase estimation.

In QPE, we need to perform Hamiltonian simulation, i.e., e^{iHt}. It is a unitary when H is Hermitian.

Basic idea of generalising QPE

Use a similar idea to quantum phase estimation.

In QPE, we need to perform Hamiltonian simulation, i.e., e^{iHt}. It is a unitary when H is Hermitian.

For GEP $A|u\rangle = \lambda B|u\rangle$, we usually cannot do $e^{iB^{-1}At}$ directly.
$\rightarrow$ not unitary, hard to compute $B^{-1}A$.

Basic idea of generalising QPE

Use a similar idea to quantum phase estimation.

In QPE, we need to perform Hamiltonian simulation, i.e., e^{iHt}. It is a unitary when H is Hermitian.

For GEP $A|u\rangle = \lambda B|u\rangle$, we usually cannot do $e^{iB^{-1}At}$ directly.
$\rightarrow$ not unitary, hard to compute $B^{-1}A$.

But we know e^{iHt} comes from the simulation of Schrödinger equation on a quantum computer:

$$i\hbar\frac{d}{dt}|\Psi(t)\rangle = H|\Psi(t)\rangle.$$

Basic idea of generalising QPE

So for general A, if we want to compute e^{iAt} we can consider the solving of ODEs on a quantum computer

$$\frac{d}{dt}|x(t)\rangle = iA|x(t)\rangle.$$

Basic idea of generalising QPE

So for general A, if we want to compute e^{iAt} we can consider the solving of ODEs on a quantum computer

$$\frac{d}{dt}|x(t)\rangle = iA|x(t)\rangle.$$

For GEP, we can consider

$$B\frac{d}{dt}|x(t)\rangle = iA|x(t)\rangle.$$

Basic idea of generalising QPE

So for general A, if we want to compute e^{iAt} we can consider the solving of ODEs on a quantum computer

$$\frac{d}{dt}|x(t)\rangle = iA|x(t)\rangle.$$

For GEP, we can consider

$$B\frac{d}{dt}|x(t)\rangle = iA|x(t)\rangle.$$

Good thing: we know how to solve ODEs on a quantum computer.

[Berry, arXiv:1010.2745]
[Berry, Childs, Ostrander, Wang, arXiv:1701.03684]
[Childs, Liu, arXiv:1901.00961]

A new way of viewing QPE

Consider the ODE $x(t)' = iAx(t)$. Suppose we use the finite difference method. Let

$$t_l = lh, \quad l = 0, 1, \ldots, m, \quad h = \frac{1}{m}.$$

A new way of viewing QPE

Consider the ODE $x(t)' = iAx(t)$. Suppose we use the finite difference method. Let

$$t_l = lh, \quad l = 0, 1, \ldots, m, \quad h = \frac{1}{m}.$$

Then

$$iAx(t_l) = \frac{d}{dt}x(t_l) \approx \frac{x(t_l) - x(t_{l-1})}{h}, \quad l = 1, \ldots, m.$$

A new way of viewing QPE

Consider the ODE $x(t)' = iAx(t)$. Suppose we use the finite difference method. Let

$$t_l = lh, \quad l = 0, 1, \ldots, m, \quad h = \frac{1}{m}.$$

Then

$$iAx(t_l) = \frac{d}{dt}x(t_l) \approx \frac{x(t_l) - x(t_{l-1})}{h}, \quad l = 1, \ldots, m.$$

At the end, we obtain a linear system

$$\begin{pmatrix} ihA - I & & & \\ I & ihA - I & & \\ & \ddots & \ddots & \\ & & I & ihA - I \end{pmatrix} \begin{pmatrix} x(h) \\ x(2h) \\ \vdots \\ x(mh) \end{pmatrix} = \begin{pmatrix} -x(0) \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

A new way of viewing QPE

The solution is

$$\sum_{j=1}^{m} |j\rangle |x(jh)\rangle = \sum_{j=1}^{m} |j\rangle \otimes e^{iAjh} |x(0)\rangle.$$

A new way of viewing QPE

The solution is

$$\sum_{j=1}^{m} |j\rangle|x(jh)\rangle = \sum_{j=1}^{m} |j\rangle \otimes e^{iAjh}|x(0)\rangle.$$

Suppose

$$|x(0)\rangle = \sum_{k=1}^{n} \beta_k |u_k\rangle,$$

then the solution is

$$\sum_{j=1}^{m}\sum_{k=1}^{n} \beta_k |j\rangle \otimes e^{iAjh}|u_k\rangle = \sum_{j=1}^{m}\sum_{k=1}^{n} \beta_k e^{i\lambda_k jh}|j\rangle \otimes |u_k\rangle.$$

A new way of viewing QPE

The solution is

$$\sum_{j=1}^{m} |j\rangle|x(jh)\rangle = \sum_{j=1}^{m} |j\rangle \otimes e^{iAjh}|x(0)\rangle.$$

Suppose

$$|x(0)\rangle = \sum_{k=1}^{n} \beta_k |u_k\rangle,$$

then the solution is

$$\sum_{j=1}^{m}\sum_{k=1}^{n} \beta_k |j\rangle \otimes e^{iAjh}|u_k\rangle = \sum_{j=1}^{m}\sum_{k=1}^{n} \beta_k e^{i\lambda_k jh}|j\rangle \otimes |u_k\rangle.$$

This is exactly the state we need in QPE by using Hamiltonian simulation. Now we just apply quantum Fourier transform to $|j\rangle$.

Some comments

We do not need to solve the ODE completely. We only care about the superposition of the discretised solutions.

The deduced linear system has certain structures so that quantum linear solver is efficient. The hard part is the complexity analysis.

This reduces the problem of solving eigenvalues to the problem of solving ODEs on a quantum computer.

Generalised eigenvalue problems

For generalised eigenvalue problem, we consider

$$B\frac{d}{dt}|x(t)\rangle = iA|x(t)\rangle.$$

But finite difference method is less efficient, we need to construct efficient quantum algorithms for solving ODEs.

Previous quantum algorithms for ODEs are not applicable here.
$\rightarrow$ Try Fourier spectral method.

Fourier spectral method

For any function $f(t)$, where $t \in [0, L]$, the K-truncated Fourier series of $f(t)$ is defined by

$$f_K(t) = \sum_{k=-K}^{K} c_k e^{2\pi i k t/L} = \sum_{k=0}^{2K} c_{k-K} e^{2\pi i (k-K) t/L},$$

where

$$c_k = \frac{1}{L} \int_0^L f(t) e^{-2\pi i k t/L} dt.$$

Fourier spectral method

For any function $f(t)$, where $t \in [0, L]$, the K-truncated Fourier series of $f(t)$ is defined by

$$f_K(t) = \sum_{k=-K}^{K} c_k e^{2\pi i k t/L} = \sum_{k=0}^{2K} c_{k-K} e^{2\pi i (k-K) t/L},$$

where

$$c_k = \frac{1}{L} \int_0^L f(t) e^{-2\pi i k t/L} dt.$$

So we can represent the j-th entry as

$$x_j(lh) = \sum_{k=0}^{2K} c_{jk} e^{2\pi i (k-K) \frac{lh}{L}},$$

where c_{jk} are the unknown Fourier coefficients we aim to compute.

Fourier spectral method

At the end, we obtain a large linear system of c_{jk}. In the quantum case, apply a quantum linear solver, we obtain its quantum state

$$\sum_{j,k} c_{j,k}|j\rangle|k\rangle.$$

Fourier spectral method

At the end, we obtain a large linear system of c_{jk}. In the quantum case, apply a quantum linear solver, we obtain its quantum state

$$\sum_{j,k} c_{j,k}|j\rangle|k\rangle.$$

What we want is the state

$$\sum_{l} |x(lh)\rangle|l\rangle = \sum_{l}\sum_{j,k} c_{jk} e^{2\pi i(k-K)\frac{lh}{L}} |j\rangle|l\rangle.$$

We only need to apply a unitary to $|k\rangle$.

The overall complexity

Suppose we have block-encodings of A, B

$$U_A = \begin{pmatrix} A/\alpha_A & \cdot \\ \cdot & \cdot \end{pmatrix}, \quad U_B = \begin{pmatrix} B/\alpha_B & \cdot \\ \cdot & \cdot \end{pmatrix},$$

then the complexity is

$$\widetilde{O}\left(\frac{(\alpha_A + \alpha_B)\kappa_B\kappa_E}{\varepsilon}\right),$$

where

- κ_E is the condition number of the matrix of eigenvectors
- κ_B is the condition number of B

The overall complexity

Regarding the special case: A is symmetric and B is positive semi-definite.

The cost of QPE method is

$$\widetilde{O}\left(\frac{(\alpha_A+\alpha_B)\kappa_B^2}{\varepsilon}\right).$$

The cost of ODE method is

$$\widetilde{O}\left(\frac{(\alpha_A+\alpha_B)\kappa_B^{1.5}}{\varepsilon}\right).$$

Both have the optimal dependence on ε. The ODE method is more general.

Polynomial eigenvalue problems (brief)

Given matrices $A_m, \cdots, A_1, A_0$, determine those values of λ and those vectors $\mathbf{x}$ for which

$$(\lambda^m A_m + \cdots + \lambda A_1 + A_0)\mathbf{x} = 0.$$

Polynomial eigenvalue problems (brief)

Given matrices $A_m, \cdots, A_1, A_0$, determine those values of λ and those vectors $\mathbf{x}$ for which

$$(\lambda^m A_m + \cdots + \lambda A_1 + A_0)\mathbf{x} = 0.$$

This is closely related to the higher order ODE

$$\left(A_m \frac{d^m}{dt^m} + \cdots + A_1 \frac{d}{dt} + A_0 \right) \mathbf{x}(t) = 0.$$

Polynomial eigenvalue problems (brief)

Given matrices $A_m, \cdots, A_1, A_0$, determine those values of λ and those vectors $\mathbf{x}$ for which

$$(\lambda^m A_m + \cdots + \lambda A_1 + A_0)\mathbf{x} = 0.$$

This is closely related to the higher order ODE

$$\left(A_m \frac{d^m}{dt^m} + \cdots + A_1 \frac{d}{dt} + A_0 \right) \mathbf{x}(t) = 0.$$

If the initial state is a linear combination of eigenvectors

$$\mathbf{x}(0) = \sum_j \beta_j |u_j\rangle,$$

then the solution is

$$\mathbf{x}(t) = \sum_j \beta_j e^{\lambda_j t} |u_j\rangle.$$

More on the assumptions

Recall that we assumed

- B is invertible
- $B^{-1}A$ is diagonalizable
- all eigenvalues are real

Regarding the first one, we have the following hardness result.

> If $A|\psi\rangle = \lambda B|\psi\rangle$ and B is singular, then finding λ requires at least $\Omega(\sqrt{n})$ quantum queries to A and B.

More on the assumptions

Recall that we assumed

- B is invertible
- $B^{-1}A$ is diagonalizable
- all eigenvalues are real

Regarding the third one, if we can start from this state

$$\sum_{j=1}^{n} \beta_j |u_j\rangle |\bar{u}_j\rangle,$$

then we can solve the complex case by approximating the real and imaginary parts separately. But this state is hard to prepare.
$\rightarrow$ How to generalise this?

More on the assumptions

Recall that we assumed

- B is invertible
- $B^{-1}A$ is diagonalizable
- all eigenvalues are real

Regarding the third one, if we can start from this state

$$\sum_{j=1}^{n} \beta_j |u_j\rangle |\bar{u}_j\rangle,$$

then we can solve the complex case by approximating the real and imaginary parts separately. But this state is hard to prepare.
$\rightarrow$ How to generalise this?

Thanks very much for your attention!