

Quantum and Classical Query Complexities of Functions of Matrices

Changpeng Shao

Academy of Mathematics and Systems Science, Chinese Academy of Sciences
based on joint work with **Ashley Montanaro** (University of Bristol & Phasecraft)
[arXiv:2311.06999 \(STOC 2024\)](#)

QuSoft Seminar
7 March 2025

Outline of the talk

- ▶ Background and motivations
- ▶ Lower bounds of query complexity
- ▶ BQP-completeness
- ▶ Conclusions

Background and motivations

Lower bounds of query complexity

BQP-completeness

Conclusion

Quantum linear algebra

- ▶ Quantum linear algebra plays an important role in the exploration of quantum advantages.

Quantum linear algebra

- ▶ Quantum linear algebra plays an important role in the exploration of quantum advantages.
- ▶ Hamiltonian simulation: **computing** $e^{iAt}|\mathbf{b}\rangle$, where A is Hermitian. A fundamentally important problem in quantum computing.

Quantum linear algebra

- ▶ Quantum linear algebra plays an important role in the exploration of quantum advantages.
- ▶ Hamiltonian simulation: **computing** $e^{iAt}|\mathbf{b}\rangle$, where A is Hermitian. A fundamentally important problem in quantum computing.
- ▶ The HHL algorithm for linear systems: **computing** $A^{-1}|\mathbf{b}\rangle$. Greatly promoted the development of quantum linear algebra [[Harrow-Hassidim-Lloyd, '08](#)].

Quantum linear algebra

- ▶ Quantum linear algebra plays an important role in the exploration of quantum advantages.
- ▶ Hamiltonian simulation: **computing** $e^{iAt}|\mathbf{b}\rangle$, where A is Hermitian. A fundamentally important problem in quantum computing.
- ▶ The HHL algorithm for linear systems: **computing** $A^{-1}|\mathbf{b}\rangle$. Greatly promoted the development of quantum linear algebra [Harrow-Hassidim-Lloyd, '08].
- ▶ Has wide applications, such as
 - ▶ Quantum recommendation systems: **computing** $A_{\geq\delta}|i\rangle$, where $A_{\geq\delta}$ is the truncation by keeping singular values larger than δ [Kerenidis-Prakash, '16].
 - ▶ Solving linear differential equations: **computing** $e^{At}|\mathbf{b}\rangle$ [Berry, '10].

Quantum linear algebra

- ▶ Quantum linear algebra plays an important role in the exploration of quantum advantages.
- ▶ Hamiltonian simulation: **computing** $e^{iAt}|\mathbf{b}\rangle$, where A is Hermitian. A fundamentally important problem in quantum computing.
- ▶ The HHL algorithm for linear systems: **computing** $A^{-1}|\mathbf{b}\rangle$. Greatly promoted the development of quantum linear algebra [Harrow-Hassidim-Lloyd, '08].
- ▶ Has wide applications, such as
 - ▶ Quantum recommendation systems: **computing** $A_{\geq\delta}|i\rangle$, where $A_{\geq\delta}$ is the truncation by keeping singular values larger than δ [Kerenidis-Prakash, '16].
 - ▶ Solving linear differential equations: **computing** $e^{At}|\mathbf{b}\rangle$ [Berry, '10].
- ▶ **All these problems can be described as functions of matrices: computing $f(A)|\mathbf{b}\rangle$.**

Quantum linear algebra

- ▶ Quantum linear algebra plays an important role in the exploration of quantum advantages.
- ▶ Hamiltonian simulation: **computing** $e^{iAt}|\mathbf{b}\rangle$, where A is Hermitian. A fundamentally important problem in quantum computing.
- ▶ The HHL algorithm for linear systems: **computing** $A^{-1}|\mathbf{b}\rangle$. Greatly promoted the development of quantum linear algebra [Harrow-Hassidim-Lloyd, '08].
- ▶ Has wide applications, such as
 - ▶ Quantum recommendation systems: **computing** $A_{\geq\delta}|i\rangle$, where $A_{\geq\delta}$ is the truncation by keeping singular values larger than δ [Kerenidis-Prakash, '16].
 - ▶ Solving linear differential equations: **computing** $e^{At}|\mathbf{b}\rangle$ [Berry, '10].
- ▶ **All these problems can be described as functions of matrices: computing** $f(A)|\mathbf{b}\rangle$.
- ▶ Can be solved by a similar idea to HHL, but more efficiently by quantum singular value transform [Gilyén-Su-Low-Wiebe, '18].

Quantum singular value transform (QSVT)

Assume A is Hermitian and $\|A\| \leq 1$ for simplicity.

Quantum singular value transform (QSVT)

Assume A is Hermitian and $\|A\| \leq 1$ for simplicity.

Given a unitary of the form

$$U = \begin{pmatrix} A & * \\ * & * \end{pmatrix},$$

for any **1-bounded polynomial** $f(x)$ of **degree** d , there is a quantum circuit that implements a unitary

$$\tilde{U} = \begin{pmatrix} f(A) & * \\ * & * \end{pmatrix}.$$

Moreover, the circuit uses U, U^\dagger and the controlled forms $O(d)$ times in total.

Quantum singular value transform (QSVT)

Assume A is Hermitian and $\|A\| \leq 1$ for simplicity.

Given a unitary of the form

$$U = \begin{pmatrix} A & * \\ * & * \end{pmatrix},$$

for any **1-bounded polynomial** $f(x)$ of **degree** d , there is a quantum circuit that implements a unitary

$$\tilde{U} = \begin{pmatrix} f(A) & * \\ * & * \end{pmatrix}.$$

Moreover, the circuit uses U, U^\dagger and the controlled forms $O(d)$ times in total.

As a direct result, we can compute $f(A)|\mathbf{b}\rangle$.

Quantum singular value transform (QSVT)

Assume A is Hermitian and $\|A\| \leq 1$ for simplicity.

Given a unitary of the form

$$U = \begin{pmatrix} A & * \\ * & * \end{pmatrix},$$

for any **1-bounded polynomial** $f(x)$ of **degree** d , there is a quantum circuit that implements a unitary

$$\tilde{U} = \begin{pmatrix} f(A) & * \\ * & * \end{pmatrix}.$$

Moreover, the circuit uses U, U^\dagger and the controlled forms $O(d)$ times in total.

As a direct result, we can compute $f(A)|\mathbf{b}\rangle$.

Still works if f is not a polynomial, just consider its polynomial approximation.

This talk

Motivations: For functions of matrices,

- ▶ QSVT is optimal in many cases, how about the general case?
- ▶ What is the quantum-classical separation?

This talk

Motivations: For functions of matrices,

- ▶ QSVT is optimal in many cases, how about the general case?
- ▶ What is the quantum-classical separation?

To better understand the above questions, we consider the following weaker problem

Problem (Approximate an entry of $f(A)$)

Let $f(x) : [-1, 1] \rightarrow [-1, 1]$ be a function, let A be *sparse* and *Hermitian* with $\|A\| \leq 1$. Given two indices i, j and accuracy ε , compute $\langle i | f(A) | j \rangle \pm \varepsilon$.

If $A = UDU^{-1}$ is the eigenvalue decomposition, then $f(A) := Uf(D)U^{-1}$, where f is applied to the eigenvalues.

A is called s -sparse if the number of nonzero entries in each row/column is at most s .

Query complexity

For a sparse matrix $A = (A_{i,j})$, we are given 2 (commonly used) oracles:

$$\begin{aligned}(i, j) &\longrightarrow \mathcal{O}_1 \longrightarrow p_{i,j} \\(i, j) &\longrightarrow \mathcal{O}_2 \longrightarrow A_{i,j}\end{aligned}$$

where p_{ij} is the index of the j -th nonzero entry in the i -th row.

In the quantum case, we assume the oracles can be used in superposition, e.g.,
 $\sum \alpha_{i,j} |i, j\rangle |0\rangle \mapsto \sum \alpha_{i,j} |i, j\rangle |A_{i,j}\rangle$

The **query complexity** is the minimal number of calls to the oracles to solve the problem.

Background and motivations

Lower bounds of query complexity

BQP-completeness

Conclusion

An example: matrix powers A^m

Classical algorithms:

- ▶ Assume A is s -sparse, then by definition

$$(A^m)_{i,j} = \sum_{k_1} \sum_{k_2} \cdots \sum_{k_{d-1}} A_{i,k_1} A_{k_1,k_2} \cdots A_{k_{m-1},j}$$

The complexity is $O(s^{m-1})$

An example: matrix powers A^m

Classical algorithms:

- ▶ Assume A is s -sparse, then by definition

$$(A^m)_{i,j} = \sum_{k_1} \sum_{k_2} \cdots \sum_{k_{d-1}} A_{i,k_1} A_{k_1,k_2} \cdots A_{k_{d-1},j}$$

The complexity is $O(s^{m-1})$

- ▶ For x^m , there is an approximation polynomial of degree $\Theta(\sqrt{m})$, so can be improved to $s^{\tilde{O}(\sqrt{m})}$ [Sachdeva & Vishnoi, 2014]

An example: matrix powers A^m

Classical algorithms:

- ▶ Assume A is s -sparse, then by definition

$$(A^m)_{i,j} = \sum_{k_1} \sum_{k_2} \cdots \sum_{k_{d-1}} A_{i,k_1} A_{k_1,k_2} \cdots A_{k_{d-1},j}$$

The complexity is $O(s^{m-1})$

- ▶ For x^m , there is an approximation polynomial of degree $\Theta(\sqrt{m})$, so can be improved to $s^{\tilde{O}(\sqrt{m})}$ [Sachdeva & Vishnoi, 2014]
- ▶ The problem is known to be **BQP-complete** [Janzing & Wocjan, 2007]

An example: matrix powers A^m

Classical algorithms:

- ▶ Assume A is s -sparse, then by definition

$$(A^m)_{i,j} = \sum_{k_1} \sum_{k_2} \cdots \sum_{k_{d-1}} A_{i,k_1} A_{k_1,k_2} \cdots A_{k_{d-1},j}$$

The complexity is $O(s^{m-1})$

- ▶ For x^m , there is an approximation polynomial of degree $\Theta(\sqrt{m})$, so can be improved to $s^{\tilde{O}(\sqrt{m})}$ [Sachdeva & Vishnoi, 2014]
- ▶ The problem is known to be **BQP-complete** [Janzing & Wocjan, 2007]
- ▶ Our result: $\tilde{\Omega}((s/2)^{(\sqrt{m}-1)/6})$

An example: matrix powers A^m

Classical algorithms:

- ▶ Assume A is s -sparse, then by definition

$$(A^m)_{i,j} = \sum_{k_1} \sum_{k_2} \cdots \sum_{k_{d-1}} A_{i,k_1} A_{k_1,k_2} \cdots A_{k_{d-1},j}$$

The complexity is $O(s^{m-1})$

- ▶ For x^m , there is an approximation polynomial of degree $\Theta(\sqrt{m})$, so can be improved to $s^{\tilde{O}(\sqrt{m})}$ [Sachdeva & Vishnoi, 2014]
- ▶ The problem is known to be **BQP-complete** [Janzing & Wocjan, 2007]
- ▶ Our result: $\tilde{\Omega}((s/2)^{(\sqrt{m}-1)/6})$

Quantum algorithms:

- ▶ Upper bound by QSVT $O(s\sqrt{m}/\varepsilon)$

An example: matrix powers A^m

Classical algorithms:

- ▶ Assume A is s -sparse, then by definition

$$(A^m)_{i,j} = \sum_{k_1} \sum_{k_2} \cdots \sum_{k_{m-1}} A_{i,k_1} A_{k_1,k_2} \cdots A_{k_{m-1},j}$$

The complexity is $O(s^{m-1})$

- ▶ For x^m , there is an approximation polynomial of degree $\Theta(\sqrt{m})$, so can be improved to $s^{\tilde{O}(\sqrt{m})}$ [Sachdeva & Vishnoi, 2014]
- ▶ The problem is known to be **BQP-complete** [Janzing & Wocjan, 2007]
- ▶ Our result: $\tilde{\Omega}((s/2)^{(\sqrt{m}-1)/6})$

Quantum algorithms:

- ▶ Upper bound by QSVT $O(s\sqrt{m}/\varepsilon)$
- ▶ Our result (lower bound): $\Omega(\sqrt{m})$

General case (our results)

Assume f is continuous, A is s -sparse and Hermitian, then computing $f(A)_{i,j} \pm \varepsilon$ costs

	Quantum algorithm	Classical algorithm
Upper bound	$O(sd/\varepsilon)$	$O(s^{d-1})$
Lower bound	$\Omega(d)$	$\tilde{\Omega}((s/2)^{(d-1)/6})$

where $d = \widetilde{\deg}_\varepsilon(f)$ is the **approximate degree**:

$$\begin{aligned}\widetilde{\deg}_\varepsilon(f) = \min\{d : |f(x) - g(x)| \leq \varepsilon, \forall x \in [-1, 1], \\ g(x) \text{ is a polynomial of degree } d\}.\end{aligned}$$

The quantum lower bound is similar to the famous polynomial method for Boolean functions [Beals, Buhrman, Cleve, Mosca, de Wolf, FOCS '98].

Key theorem in the proofs

Theorem (Key theorem)

Let $f : [-1, 1] \rightarrow [-1, 1]$ be continuous with odd part $f_{\text{odd}}(x) = \frac{f(x) - f(-x)}{2}$, then there is a **symmetric tridiagonal matrix**

$$A = \begin{pmatrix} 0 & b_1 & & & \\ b_1 & 0 & b_2 & & \\ & b_2 & \ddots & \ddots & \\ & & \ddots & \ddots & b_{n-1} \\ & & & b_{n-1} & 0 \end{pmatrix}_{n \times n}$$

satisfying $b_i \neq 0$, $\|A\| \leq 1$ and $f(A)_{1,n} \geq \varepsilon$, where $n = \widetilde{\deg_\varepsilon}(f_{\text{odd}}) + O(1)$.

Proof. linear semi-infinite programming + dual polynomial method + properties of tridiagonal matrices.

Lower bound's proof of quantum algorithms

Parity problem: Given $x_1, \dots, x_n \in \{0, 1\}$, compute $x_1 \oplus \dots \oplus x_n$, the quantum query complexity is $\Theta(n)$

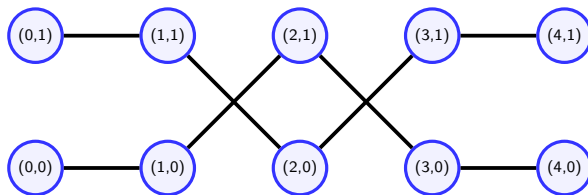
Lower bound's proof of quantum algorithms

Parity problem: Given $x_1, \dots, x_n \in \{0, 1\}$, compute $x_1 \oplus \dots \oplus x_n$, the quantum query complexity is $\Theta(n)$

We construct a weighted graph G :

- ▶ **vertices:** (i, b) , where $i \in \{0, 1, \dots, n\}, b \in \{0, 1\}$
- ▶ **edges:** an edge between $(i-1, b)$ and $(i, b \oplus x_i)$
- ▶ **weights:** chosen carefully

For example, $(x_1, x_2, x_3, x_4) = (0, 1, 1, 0)$, then G is



Lower bound's proof of quantum algorithms

Essentially, G consists of two paths

$$(0, 0) - (1, x_1) - (2, x_1 \oplus x_2) - \cdots - (n, x_1 \oplus \cdots \oplus x_n)$$

$$(0, 1) - (1, 1 \oplus x_1) - (2, 1 \oplus x_1 \oplus x_2) - \cdots - (n, 1 \oplus x_1 \oplus \cdots \oplus x_n)$$

Lower bound's proof of quantum algorithms

Essentially, G consists of two paths

$$(0, 0) - (1, x_1) - (2, x_1 \oplus x_2) - \cdots - (n, x_1 \oplus \cdots \oplus x_n)$$

$$(0, 1) - (1, 1 \oplus x_1) - (2, 1 \oplus x_1 \oplus x_2) - \cdots - (n, 1 \oplus x_1 \oplus \cdots \oplus x_n)$$

Let A be the adjacency matrix of G (essentially two symmetric tridiagonal matrices).

► **Case 1:** if $x_1 \oplus x_2 \oplus \cdots \oplus x_n = 0$, then $\langle 0, 0 | f(A) | n, 1 \rangle = 0$

Lower bound's proof of quantum algorithms

Essentially, G consists of two paths

$$(0, 0) - (1, x_1) - (2, x_1 \oplus x_2) - \cdots - (n, x_1 \oplus \cdots \oplus x_n)$$

$$(0, 1) - (1, 1 \oplus x_1) - (2, 1 \oplus x_1 \oplus x_2) - \cdots - (n, 1 \oplus x_1 \oplus \cdots \oplus x_n)$$

Let A be the adjacency matrix of G (essentially two symmetric tridiagonal matrices).

- ▶ **Case 1:** if $x_1 \oplus x_2 \oplus \cdots \oplus x_n = 0$, then $\langle 0, 0 | f(A) | n, 1 \rangle = 0$
- ▶ **Case 2:** if $x_1 \oplus x_2 \oplus \cdots \oplus x_n = 1$, then we hope to find appropriate weights such that $\langle 0, 0 | f(A) | n, 1 \rangle \geq \varepsilon$ for some ε . The weights are determined by our key theorem.

Lower bound's proof of quantum algorithms

Essentially, G consists of two paths

$$(0, 0) - (1, x_1) - (2, x_1 \oplus x_2) - \cdots - (n, x_1 \oplus \cdots \oplus x_n)$$

$$(0, 1) - (1, 1 \oplus x_1) - (2, 1 \oplus x_1 \oplus x_2) - \cdots - (n, 1 \oplus x_1 \oplus \cdots \oplus x_n)$$

Let A be the adjacency matrix of G (essentially two symmetric tridiagonal matrices).

- ▶ **Case 1:** if $x_1 \oplus x_2 \oplus \cdots \oplus x_n = 0$, then $\langle 0, 0 | f(A) | n, 1 \rangle = 0$
- ▶ **Case 2:** if $x_1 \oplus x_2 \oplus \cdots \oplus x_n = 1$, then we hope to find appropriate weights such that $\langle 0, 0 | f(A) | n, 1 \rangle \geq \varepsilon$ for some ε . The weights are determined by our key theorem.

As a result, if we can estimate $\langle 0, 0 | f(A) | n, 1 \rangle$, we then can solve the parity problem, which is known hard.

Lower bound's proof of quantum algorithms

Essentially, G consists of two paths

$$(0, 0) - (1, x_1) - (2, x_1 \oplus x_2) - \cdots - (n, x_1 \oplus \cdots \oplus x_n)$$

$$(0, 1) - (1, 1 \oplus x_1) - (2, 1 \oplus x_1 \oplus x_2) - \cdots - (n, 1 \oplus x_1 \oplus \cdots \oplus x_n)$$

Let A be the adjacency matrix of G (essentially two symmetric tridiagonal matrices).

- ▶ **Case 1:** if $x_1 \oplus x_2 \oplus \cdots \oplus x_n = 0$, then $\langle 0, 0 | f(A) | n, 1 \rangle = 0$
- ▶ **Case 2:** if $x_1 \oplus x_2 \oplus \cdots \oplus x_n = 1$, then we hope to find appropriate weights such that $\langle 0, 0 | f(A) | n, 1 \rangle \geq \varepsilon$ for some ε . The weights are determined by our key theorem.

As a result, if we can estimate $\langle 0, 0 | f(A) | n, 1 \rangle$, we then can solve the parity problem, which is known hard. It is very important to ensure that $n = \widetilde{\deg}(f) + O(1)$ in the key theorem.

Lower bound's proof of quantum algorithms

Essentially, G consists of two paths

$$(0, 0) - (1, x_1) - (2, x_1 \oplus x_2) - \cdots - (n, x_1 \oplus \cdots \oplus x_n)$$

$$(0, 1) - (1, 1 \oplus x_1) - (2, 1 \oplus x_1 \oplus x_2) - \cdots - (n, 1 \oplus x_1 \oplus \cdots \oplus x_n)$$

Let A be the adjacency matrix of G (essentially two symmetric tridiagonal matrices).

- ▶ **Case 1:** if $x_1 \oplus x_2 \oplus \cdots \oplus x_n = 0$, then $\langle 0, 0 | f(A) | n, 1 \rangle = 0$
- ▶ **Case 2:** if $x_1 \oplus x_2 \oplus \cdots \oplus x_n = 1$, then we hope to find appropriate weights such that $\langle 0, 0 | f(A) | n, 1 \rangle \geq \varepsilon$ for some ε . The weights are determined by our key theorem.

As a result, if we can estimate $\langle 0, 0 | f(A) | n, 1 \rangle$, we then can solve the parity problem, which is known hard. It is very important to ensure that $n = \widetilde{\deg}(f) + O(1)$ in the key theorem. This idea is inspired by the no fast-forwarding theorem. [Berry, Ahokas, Cleve, Sanders, Comm. Math. Phys. '07]

Lower bound's proof of classical algorithms

Forrelation problem (Aaronson & Ambainis, 2015):

Given $g_1, g_2 : \{0, 1\}^n \rightarrow \{\pm 1\}$, let $D_i = \text{diag}(g_i(x) : x \in \{0, 1\}^n)$, $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$,
define

$$\begin{aligned}\Phi(g_1, g_2) &:= \langle 0^n | H^{\otimes n} D_1 H^{\otimes n} D_2 H^{\otimes n} | 0^n \rangle \\ &= \frac{1}{2^{3n/2}} \sum_{x, y \in \{0, 1\}^n} (-1)^{x \cdot y} g_1(x) g_2(y).\end{aligned}$$

The goal is to compute $\Phi(g_1, g_2) \pm 1/3$

Lower bound's proof of classical algorithms

Forrelation problem (Aaronson & Ambainis, 2015):

Given $g_1, g_2 : \{0, 1\}^n \rightarrow \{\pm 1\}$, let $D_i = \text{diag}(g_i(x) : x \in \{0, 1\}^n)$, $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$,
define

$$\begin{aligned}\Phi(g_1, g_2) &:= \langle 0^n | H^{\otimes n} D_1 H^{\otimes n} D_2 H^{\otimes n} | 0^n \rangle \\ &= \frac{1}{2^{3n/2}} \sum_{x, y \in \{0, 1\}^n} (-1)^{x \cdot y} g_1(x) g_2(y).\end{aligned}$$

The goal is to compute $\Phi(g_1, g_2) \pm 1/3$

For this problem, the classical query complexity is lower bounded by $\Omega(\sqrt{2^n}/n)$, while the quantum query complexity is 1.

Feynman's clock construction

Let $U = U_{N-1} \cdots U_2 U_1$ be a unitary operator, define

$$A = \begin{pmatrix} 0 & b_1 U_1^\dagger & & \\ b_1 U_1 & 0 & b_2 U_2^\dagger & \\ & b_2 U_2 & \ddots & \ddots \\ & & \ddots & \ddots \end{pmatrix}$$

Feynman's clock construction

Let $U = U_{N-1} \cdots U_2 U_1$ be a unitary operator, define

$$A = \begin{pmatrix} 0 & b_1 U_1^\dagger & & \\ b_1 U_1 & 0 & b_2 U_2^\dagger & \\ & b_2 U_2 & \ddots & \ddots \\ & & \ddots & \ddots \end{pmatrix}$$

Let $|\psi_t\rangle := |t\rangle \otimes U_t \cdots U_1 |0\rangle$, then

$$A|\psi_t\rangle = b_{t-1}|\psi_{t-1}\rangle + b_{t+1}|\psi_{t+1}\rangle$$

In subspace $\{|\psi_t\rangle : t = 0, 1, \dots, N-1\}$, A is a **symmetric tridiagonal matrix** of dimension N .

Lower bound's proof of classical algorithms

In the Forrelation problem, we have $U = H^{\otimes n} D_1 H^{\otimes n} D_2 H^{\otimes n}$. To ensure A is sparse in the clock construction, we decompose

$$H^{\otimes n} = (H \otimes I \otimes \cdots \otimes I)(I \otimes H \otimes \cdots \otimes I) \cdots (I \otimes I \otimes \cdots \otimes H)$$

Lower bound's proof of classical algorithms

In the Forrelation problem, we have $U = H^{\otimes n} D_1 H^{\otimes n} D_2 H^{\otimes n}$. To ensure A is sparse in the clock construction, we decompose

$$H^{\otimes n} = (H \otimes I \otimes \cdots \otimes I)(I \otimes H \otimes \cdots \otimes I) \cdots (I \otimes I \otimes \cdots \otimes H)$$

Now $N = 3n + 2$,

$$\begin{aligned} |\psi_0\rangle &= |0\rangle \otimes |0\rangle \\ |\psi_{N-1}\rangle &= |N-1\rangle \otimes H^{\otimes n} D_1 H^{\otimes n} D_2 H^{\otimes n} |0\rangle \end{aligned}$$

Lower bound's proof of classical algorithms

In the Forrelation problem, we have $U = H^{\otimes n} D_1 H^{\otimes n} D_2 H^{\otimes n}$. To ensure A is sparse in the clock construction, we decompose

$$H^{\otimes n} = (H \otimes I \otimes \cdots \otimes I)(I \otimes H \otimes \cdots \otimes I) \cdots (I \otimes I \otimes \cdots \otimes H)$$

Now $N = 3n + 2$,

$$|\psi_0\rangle = |0\rangle \otimes |0\rangle$$

$$|\psi_{N-1}\rangle = |N-1\rangle \otimes H^{\otimes n} D_1 H^{\otimes n} D_2 H^{\otimes n} |0\rangle$$

Let $|\phi_{N-1}\rangle = |N-1\rangle \otimes |0\rangle$, then

$$\langle \phi_{N-1} | f(A) | \psi_0 \rangle = \langle \psi_{N-1} | f(A) | \psi_0 \rangle \cdot \Phi(g_1, g_2)$$

Lower bound's proof of classical algorithms

In the Forrelation problem, we have $U = H^{\otimes n} D_1 H^{\otimes n} D_2 H^{\otimes n}$. To ensure A is sparse in the clock construction, we decompose

$$H^{\otimes n} = (H \otimes I \otimes \cdots \otimes I)(I \otimes H \otimes \cdots \otimes I) \cdots (I \otimes I \otimes \cdots \otimes H)$$

Now $N = 3n + 2$,

$$|\psi_0\rangle = |0\rangle \otimes |0\rangle$$

$$|\psi_{N-1}\rangle = |N-1\rangle \otimes H^{\otimes n} D_1 H^{\otimes n} D_2 H^{\otimes n} |0\rangle$$

Let $|\phi_{N-1}\rangle = |N-1\rangle \otimes |0\rangle$, then

$$\langle \phi_{N-1} | f(A) | \psi_0 \rangle = \langle \psi_{N-1} | f(A) | \psi_0 \rangle \cdot \Phi(g_1, g_2)$$

↓

an entry of $f(A)$

↓

Easy

↓

Hard

Lower bound's proof of classical algorithms

In the Forrelation problem, we have $U = H^{\otimes n} D_1 H^{\otimes n} D_2 H^{\otimes n}$. To ensure A is sparse in the clock construction, we decompose

$$H^{\otimes n} = (H \otimes I \otimes \cdots \otimes I)(I \otimes H \otimes \cdots \otimes I) \cdots (I \otimes I \otimes \cdots \otimes H)$$

Now $N = 3n + 2$,

$$|\psi_0\rangle = |0\rangle \otimes |0\rangle$$

$$|\psi_{N-1}\rangle = |N-1\rangle \otimes H^{\otimes n} D_1 H^{\otimes n} D_2 H^{\otimes n} |0\rangle$$

Let $|\phi_{N-1}\rangle = |N-1\rangle \otimes |0\rangle$, then

$$\langle \phi_{N-1} | f(A) | \psi_0 \rangle = \langle \psi_{N-1} | f(A) | \psi_0 \rangle \cdot \Phi(g_1, g_2)$$

↓

an entry of $f(A)$

↓

Hard

↓

Easy

↓

Hard

Background and motivations

Lower bounds of query complexity

BQP-completeness

Conclusion

BQP-completeness

Definition (PromiseBQP)

PromiseBQP is the set of promise problems $(\Pi_{\text{YES}}, \Pi_{\text{NO}})$ that can be solved by a uniform family of quantum circuits. Namely, there is a quantum circuit Y such that the application of Y to the computational basis state $|x, 0\rangle$ produces the state

$$Y|x, 0\rangle = \alpha_{x,0}|0\rangle \otimes |\psi_{x,0}\rangle + \alpha_{x,1}|1\rangle \otimes |\psi_{x,1}\rangle$$

such that

- ▶ *for every $x \in \Pi_{\text{YES}}$ it holds that $|\alpha_{x,1}|^2 \geq 2/3$ and*
- ▶ *for every $x \in \Pi_{\text{NO}}$ it holds that $|\alpha_{x,1}|^2 \leq 1/3$.*

Equivalently, $|\alpha_{x,1}|^2 - |\alpha_{x,0}|^2 \geq 1/3$ if $x \in \Pi_{\text{YES}}$ and $|\alpha_{x,1}|^2 - |\alpha_{x,0}|^2 \leq -1/3$ if $x \in \Pi_{\text{NO}}$.

Why we care about BQP-completeness

- ▶ It defines the hardest problems in BQP

BQP = problems can be solved efficiently on a quantum computer

- ▶ It characterizes the power of quantum computing

So we learn what quantum computers can do that classical computers may struggle with.

If a BQP-complete is outside P, it confirms that quantum computers can solve problems that classical ones cannot efficiently.

Known examples: computing e^{iAt} , A^{-1} , A^d ; approximating the Jones polynomial, etc.

Entry estimation problem

Problem

Let $f(x) : [-1, 1] \rightarrow [-1, 1]$ be a continuous function, let A be an $n \times n$ sparse Hermitian matrix such that $\|A\| \leq 1$. Let $\varepsilon_1, \varepsilon_2 \in (0, 1)$ and i, j be two indices. Assume that one of the following holds:

- ▶ YES case: if $f(A)_{ij} \geq \varepsilon_1$,
- ▶ NO case: if $f(A)_{ij} \leq -\varepsilon_2$.

Decide which is the case.

Entry estimation problem

Problem

Let $f(x) : [-1, 1] \rightarrow [-1, 1]$ be a continuous function, let A be an $n \times n$ sparse Hermitian matrix such that $\|A\| \leq 1$. Let $\varepsilon_1, \varepsilon_2 \in (0, 1)$ and i, j be two indices. Assume that one of the following holds:

- ▶ YES case: if $f(A)_{ij} \geq \varepsilon_1$,
- ▶ NO case: if $f(A)_{ij} \leq -\varepsilon_2$.

Decide which is the case.

Theorem

Assume that $\widetilde{\deg}(f) = \Omega(\text{polylog}(n))$. Then the “entry estimation problem” is PromiseBQP-complete.

Proof sketch

- ▶ The proof is already hidden in the proof of classical lower bounds

Proof sketch

- ▶ The proof is already hidden in the proof of classical lower bounds
- ▶ Recall

$$|\psi_0\rangle = |0\rangle \otimes |0\rangle, \quad |\psi_{N-1}\rangle = |N-1\rangle \otimes U|0\rangle, \quad |\phi_{N-1}\rangle = |N-1\rangle \otimes |0\rangle$$

Proof sketch

- ▶ The proof is already hidden in the proof of classical lower bounds
- ▶ Recall

$$|\psi_0\rangle = |0\rangle \otimes |0\rangle, \quad |\psi_{N-1}\rangle = |N-1\rangle \otimes U|0\rangle, \quad |\phi_{N-1}\rangle = |N-1\rangle \otimes |0\rangle$$

- ▶ Let $U = Y^\dagger Z_1 Y$, then [replace $|0\rangle$ with $|x, 0\rangle$]

$$\begin{aligned} \langle \phi_{N-1} | f(A) | \psi_0 \rangle &= \langle \psi_{N-1} | f(A) | \psi_0 \rangle \cdot \langle x, 0 | U | x, 0 \rangle \\ &= \langle \psi_{N-1} | f(A) | \psi_0 \rangle \cdot (|\alpha_{x,1}|^2 - |\alpha_{x,0}|^2) \\ &= \begin{cases} \geq \varepsilon/3 & \text{if } x \in \text{YES} \\ \leq -1/3 & \text{if } x \in \text{NO} \end{cases} \end{aligned}$$

because $\langle \psi_{N-1} | f(A) | \psi_0 \rangle \in [\varepsilon, 1]$.

Background and motivations

Lower bounds of query complexity

BQP-completeness

Conclusion

Conclusion

- ▶ For functions of matrices, we proved

	Quantum algorithm	Classical algorithm
Upper bound	$O(sd/\varepsilon)$	$O(s^{d-1})$
Lower bound	$\Omega(d)$	$\Omega((s/2)^{(d-1)/6})$

- ▶ From the point of approximate degree,
 - ▶ quantum algorithm is optimal
 - ▶ quantum-classical separation is exponential
- ▶ Estimating an entry of functions of matrices is BQP-complete.

Conclusion

- ▶ For functions of matrices, we proved

	Quantum algorithm	Classical algorithm
Upper bound	$O(sd/\varepsilon)$	$O(s^{d-1})$
Lower bound	$\Omega(d)$	$\Omega((s/2)^{(d-1)/6})$

- ▶ From the point of approximate degree,
 - ▶ quantum algorithm is optimal
 - ▶ quantum-classical separation is exponential
- ▶ Estimating an entry of functions of matrices is BQP-complete.

Open questions:

- ▶ Lower bound in terms of ε in the quantum case? $\Omega(1/\varepsilon)$?
- ▶ Any applications to quantum PCP? [Gharibian, Le Gall, Dequantizing the Quantum Singular Value Transformation, STOC'22]

Conclusion

- ▶ For functions of matrices, we proved

	Quantum algorithm	Classical algorithm
Upper bound	$O(sd/\varepsilon)$	$O(s^{d-1})$
Lower bound	$\Omega(d)$	$\Omega((s/2)^{(d-1)/6})$

- ▶ From the point of approximate degree,
 - ▶ quantum algorithm is optimal
 - ▶ quantum-classical separation is exponential
- ▶ Estimating an entry of functions of matrices is BQP-complete.

Open questions:

- ▶ Lower bound in terms of ε in the quantum case? $\Omega(1/\varepsilon)$?
- ▶ Any applications to quantum PCP? [Gharibian, Le Gall, Dequantizing the Quantum Singular Value Transformation, STOC'22]

Thanks very much for your time!