Optimal quantum adversary lower bounds for ordered search

Andrew Childs
University of Waterloo

Troy Lee
Rutgers University

A mathematical question

A mathematical question

Given a matrix

$$A_n = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 & \dots & a_{n-1} \\ a_1 & a_2 & a_3 & \dots & a_{n-1} & 0 \\ a_2 & a_3 & \dots & a_{n-1} & 0 & 0 \\ \vdots & \dots & & & \vdots & \vdots \\ a_{n-2} & a_{n-1} & 0 & 0 & 0 & 0 \\ a_{n-1} & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

how large can $\sum_i a_i$ be while $\|A_n\| \leq 1$?

A mathematical question

Given a matrix

$$A_n = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 & \dots & a_{n-1} \\ a_1 & a_2 & a_3 & \dots & a_{n-1} & 0 \\ a_2 & a_3 & \dots & a_{n-1} & 0 & 0 \\ \vdots & \dots & & & \vdots & \vdots \\ a_{n-2} & a_{n-1} & 0 & 0 & 0 & 0 \\ a_{n-1} & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

how large can $\sum_i a_i$ be while $\|A_n\| \leq 1$? Let $\alpha(n)$ denote this optimal value.

A good guess

The "half" Hilbert matrix

$$Z_n = \begin{pmatrix} 1 & 1/2 & 1/3 & 1/4 & \dots & 1/n \\ 1/2 & 1/3 & 1/4 & \dots & 1/n & 0 \\ 1/3 & 1/4 & \dots & 1/n & 0 & 0 \\ \vdots & \dots & & & \vdots & \vdots \\ 1/(n-1) & 1/n & 0 & 0 & 0 & 0 \\ 1/n & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Then $\sum_i a_i \approx \ln(n)$.

A good guess

The "half" Hilbert matrix

$$Z_n = \begin{pmatrix} 1 & 1/2 & 1/3 & 1/4 & \dots & 1/n \\ 1/2 & 1/3 & 1/4 & \dots & 1/n & 0 \\ 1/3 & 1/4 & \dots & 1/n & 0 & 0 \\ \vdots & \dots & & & \vdots & \vdots \\ 1/(n-1) & 1/n & 0 & 0 & 0 & 0 \\ 1/n & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Then $\sum_i a_i \approx \ln(n)$. How to upper bound $\|Z_n\|$?

Hilbert's Inequality

Consider the Hilbert matrix

$$H = \begin{pmatrix} 1 & 1/2 & 1/3 & 1/4 & \dots \\ 1/2 & 1/3 & 1/4 & \dots & \dots \\ 1/3 & 1/4 & \dots & & \dots \\ 1/4 & \dots & & & \vdots \\ \vdots & & & \vdots & \ddots \end{pmatrix}$$

Hilbert showed (with improvement by Schur) that $\|H\| \leq \pi$.

Hilbert's Inequality

Consider the Hilbert matrix

$$H = \begin{pmatrix} 1 & 1/2 & 1/3 & 1/4 & \dots \\ 1/2 & 1/3 & 1/4 & \dots & \dots \\ 1/3 & 1/4 & \dots & & \dots \\ 1/4 & \dots & & & \vdots \\ \vdots & & & \vdots & \ddots \end{pmatrix}$$

Hilbert showed (with improvement by Schur) that $\|H\| \leq \pi$. Thus the (normalized) half Hilbert matrix demonstrates $\alpha(n) \geq \frac{\ln(n)}{\pi}$.

Our main theorem

We show that the "half" Hilbert matrix gives essentially the optimal bound:

$$\alpha(n) = \frac{\ln(n)}{\pi} + \Theta(1).$$

Our main theorem

We show that the "half" Hilbert matrix gives essentially the optimal bound:

$$\alpha(n) = \frac{\ln(n)}{\pi} + \Theta(1).$$

For the case of non-negative matrices, we are able to give the exact answer:

$$\alpha^{+}(n) = \sum_{i=0}^{n-1} \left(\frac{\binom{2i}{i}}{4^i} \right)^2$$

Our main theorem

We show that the "half" Hilbert matrix gives essentially the optimal bound:

$$\alpha(n) = \frac{\ln(n)}{\pi} + \Theta(1).$$

For the case of non-negative matrices, we are able to give the exact answer:

$$\alpha^+(n) = \sum_{i=0}^{n-1} \left(\frac{\binom{2i}{i}}{4^i} \right)^2 = \frac{1}{\pi}(\ln n + \gamma + \ln 8) + O(1/n)$$

and explicit matrices which realize this bound.

Our main theorem

We show that the "half" Hilbert matrix gives essentially the optimal bound:

$$\alpha(n) = \frac{\ln(n)}{\pi} + \Theta(1).$$

For the case of non-negative matrices, we are able to give the exact answer:

$$\alpha^+(n) = \sum_{i=0}^{n-1} \left(\frac{\binom{2i}{i}}{4^i} \right)^2 = \frac{1}{\pi}(\ln n + \gamma + \ln 8) + O(1/n)$$

and explicit matrices which realize this bound.

Note that

$$\frac{\binom{2i}{i}}{4^i} \approx \frac{4^i/\sqrt{\pi i}}{4^i} = \frac{1}{\sqrt{\pi i}}.$$

Motivation: quantum query complexity

- In classical query complexity, want to compute some function $f(x)$ and have access to the input x by queries of the form $x_i =?$ Complexity is number of queries needed on worst case input.

- Model of quantum query complexity is attractive as captures many quantum algorithms
  - Grover's search algorithm,
  - Period finding of Shor's algorithm,
  - Quantum walks: element distinctness, triangle finding, matrix multiplication

- And we can also prove lower bounds!

Ordered search problem

- Complexity of finding a given item in an ordered list.
- Given an ordered list $x_1 \leq x_2 \leq \ldots \leq x_n$ want to find position of given item z.
- Ask queries of the form $x_i \geq z$?
- Equivalently can represent problem as querying bits of input and identifying first occurrence of a '1'. For $n = 4$, for example $S = \{1111, 0111, 0011, 0001\}$.

Complexity of ordered search

- Classically answer is given by binary search: $\log n$ queries.
- In quantum case, one can do better.

Complexity of ordered search

- Classically answer is given by binary search: $\log n$ queries.
- In quantum case, one can do better. But only by a constant!

Complexity of ordered search

- Classically answer is given by binary search: $\log n$ queries.
- In quantum case, one can do better. But only by a constant!
- Upper bounds: $0.526 \log n$ [FGGS99],

Complexity of ordered search

- Classically answer is given by binary search: $\log n$ queries.
- In quantum case, one can do better. But only by a constant!
- Upper bounds: $0.526 \log n$ [FGGS99], $0.439 \log n$ [BJL04],

Complexity of ordered search

- Classically answer is given by binary search: $\log n$ queries.
- In quantum case, one can do better. But only by a constant!
- Upper bounds: $0.526 \log n$ [FGGS99], $0.439 \log n$ [BJL04], $0.433 \log n$ [CLP06],

Complexity of ordered search

- Classically answer is given by binary search: $\log n$ queries.
- In quantum case, one can do better. But only by a constant!
- Upper bounds: $0.526 \log n$ [FGGS99], $0.439 \log n$ [BJL04], $0.433 \log n$ [CLP06], $0.32 \log n$ [B-OH07] (bounded-error)

Complexity of ordered search

- Classically answer is given by binary search: $\log n$ queries.
- In quantum case, one can do better. But only by a constant!
- Upper bounds: $0.526 \log n$ [FGGS99], $0.439 \log n$ [BJL04], $0.433 \log n$ [CLP06], $0.32 \log n$ [B-OH07] (bounded-error)
- Lower bounds: $\sqrt{\log n} / \log \log n$ [BW98],

Complexity of ordered search

- Classically answer is given by binary search: $\log n$ queries.
- In quantum case, one can do better. But only by a constant!
- Upper bounds: $0.526 \log n$ [FGGS99], $0.439 \log n$ [BJL04], $0.433 \log n$ [CLP06], $0.32 \log n$ [B-OH07] (bounded-error)
- Lower bounds: $\sqrt{\log n}/\log\log n$ [BW98], $\log n/\log\log n$ [FGGS98],

Complexity of ordered search

- Classically answer is given by binary search: $\log n$ queries.
- In quantum case, one can do better. But only by a constant!
- Upper bounds: $0.526 \log n$ [FGGS99], $0.439 \log n$ [BJL04], $0.433 \log n$ [CLP06], $0.32 \log n$ [B-OH07] (bounded-error)
- Lower bounds: $\sqrt{\log n}/\log\log n$ [BW98], $\log n/\log\log n$ [FGGS98], $0.0833 \log n$ [Amb99],

Complexity of ordered search

- Classically answer is given by binary search: $\log n$ queries.
- In quantum case, one can do better. But only by a constant!
- Upper bounds: $0.526 \log n$ [FGGS99], $0.439 \log n$ [BJL04], $0.433 \log n$ [CLP06], $0.32 \log n$ [B-OH07] (bounded-error)
- Lower bounds: $\sqrt{\log n}/\log\log n$ [BW98], $\log n/\log\log n$ [FGGS98], $0.0833 \log n$ [Amb99], $\frac{1}{\pi}\ln n \approx 0.221 \log n$ [HNS01]

Complexity of ordered search

- Classically answer is given by binary search: $\log n$ queries.
- In quantum case, one can do better. But only by a constant!
- Upper bounds: $0.526 \log n$ [FGGS99], $0.439 \log n$ [BJL04], $0.433 \log n$ [CLP06], $0.32 \log n$ [B-OH07] (bounded-error)
- Lower bounds: $\sqrt{\log n}/\log\log n$ [BW98], $\log n/\log\log n$ [FGGS98], $0.0833 \log n$ [Amb99], $\frac{1}{\pi}\ln n \approx 0.221 \log n$ [HNS01]
- What is this fundamental constant of quantum information?

Lower bounds: adversary method

- Main lower bound techniques: polynomial method and adversary method.
- Adversary method developed and improved in long series of works [BBBV94, Amb00, HNS01, BSS03, Amb03, LM04, Zha04, SŠ06, HLŠ07]
- Adversary bound is an optimization problem which can be written as a semidefinite program.

$$\mathrm{ADV}(f) := \max_{\Gamma} \frac{\|\Gamma\|}{\max_i \|\Gamma \circ D_i\|}$$

where $\Gamma[x, y] = 0$ if $f(x) = f(y)$ and $D_i[x, y] = 1$ if $x_i \neq y_i$ and 0 otherwise.

The Γ matrix

$$\begin{array}{c|cc|} & f^{-1}(0) & f^{-1}(1) \\ \hline f^{-1}(0) & 0 & A \\ f^{-1}(1) & A^* & 0 \\ \hline \end{array}$$

Notice that the spectral norm of Γ equals that of A.

The $\Gamma \circ D_1$ matrix

$$
\begin{array}{cc|cc|cc}
 & & \multicolumn{2}{c|}{f^{-1}(0)} & \multicolumn{2}{c}{f^{-1}(1)} \\
 & & 0x & 1x & 0y & 1y \\
\hline
f^{-1}(0) & 0x & \multicolumn{2}{c|}{\multirow{2}{*}{0}} & 0 & B \\
 & 1x & \multicolumn{2}{c|}{} & C & 0 \\
\hline
f^{-1}(1) & 0y & 0 & C^* & \multicolumn{2}{c}{\multirow{2}{*}{0}} \\
 & 1y & B^* & 0 & \multicolumn{2}{c}{} \\
\end{array}
$$

The spectral norm of $\Gamma \circ D_1$ equals $\max\{\|B\|, \|C\|\}$.

Automorphism principle

- "Whenever you have to deal with a structure endowed entity Σ try to determine its group of automorphisms . . . you can expect to gain a deep insight into the constitution of Σ in this way."

  —Hermann Weyl, *Symmetry*

Automorphism principle

- "Whenever you have to deal with a structure endowed entity Σ try to determine its group of automorphisms . . . you can expect to gain a deep insight into the constitution of Σ in this way."

  —Hermann Weyl, *Symmetry*

- Using symmetry of problem can greatly simplify search for optimal adversary matrices [HLŠ07].

- Input to ordered search (for $n = 4$) $S = \{1111, 0111, 0011, 0001\}$

Automorphism principle

- "Whenever you have to deal with a structure endowed entity Σ try to determine its group of automorphisms . . . you can expect to gain a deep insight into the constitution of Σ in this way."

  —Hermann Weyl, *Symmetry*

- Using symmetry of problem can greatly simplify search for optimal adversary matrices [HLŠ07].

- Input to ordered search (for $n = 4$) $S = \{1111, 0111, 0011, 0001\}$ Trivial automorphism group!

Automorphism principle

- [FGGS99] extend inputs "to a circle": $S' = \{11110000, 01111000, 00111100, 00011110, 00001111, 10000111, 11000011, 11100001\}$
- Now have cyclic structure, and query complexity changes by at most 1.
- Using automorphism principle, can wlog reduce computation of adversary bound to the matrix problem given at beginning of talk.

Automorphism principle

- [FGGS99] extend inputs "to a circle": $S' = \{11110000, 01111000, 00111100, 00011110, 00001111, 10000111, 11000011, 11100001\}$
- Now have cyclic structure, and query complexity changes by at most 1.
- Using automorphism principle, can wlog reduce computation of adversary bound to the matrix problem given at beginning of talk.
- We show that the adversary method (even with negative weights) cannot show lower bounds larger than $\frac{1}{\pi} \ln n + O(1)$.

A word about the proof (non-negative case)

- We exhibit solutions to both the primal and dual formulation of adversary bound, and show that they match.

- A key role in both directions is played by the lovely sequence

$$\beta_i = \frac{\binom{2i}{i}}{4^i}.$$

A word about the proof (non-negative case)

- We exhibit solutions to both the primal and dual formulation of adversary bound, and show that they match.

- A key role in both directions is played by the lovely sequence

$$\beta_i = \frac{\binom{2i}{i}}{4^i}.$$

- Key property: $\sum_{i=0}^{j} \beta_i \beta_{j-i} = 1$

A word about the proof (non-negative case)

- We exhibit solutions to both the primal and dual formulation of adversary bound, and show that they match.

- A key role in both directions is played by the lovely sequence

$$\beta_i = \frac{\binom{2i}{i}}{4^i}.$$

- Key property: $\sum_{i=0}^{j} \beta_i \beta_{j-i} = 1$

- Proof:

$$\frac{1}{\sqrt{1-z}} = \beta_0 + \beta_1 z + \beta_2 z^2 + \beta_3 z^3 + \dots$$

Optimal matrix

Recall we wish to show that $\alpha^+(n) = \sum_{i=0}^{n-1} \left(\frac{\binom{2i}{i}}{4^i} \right)^2$.

Optimal matrix

Recall we wish to show that $\alpha^+(n) = \sum_{i=0}^{n-1} \left(\frac{\binom{2i}{i}}{4^i} \right)^2$.

Define $A_n(j) = \sum_{i=0}^{n-j-1} \beta_i \beta_{i+j}$.

$$\begin{pmatrix} A_4(0) - A_4(1) & A_4(1) - A_4(2) & A_4(2) - A_4(3) & A_4(3) \\ A_4(1) - A_4(2) & A_4(2) - A_4(3) & A_4(3) & 0 \\ A_4(2) - A_4(3) & A_4(3) & 0 & 0 \\ A_4(3) & 0 & 0 & 0 \end{pmatrix}$$

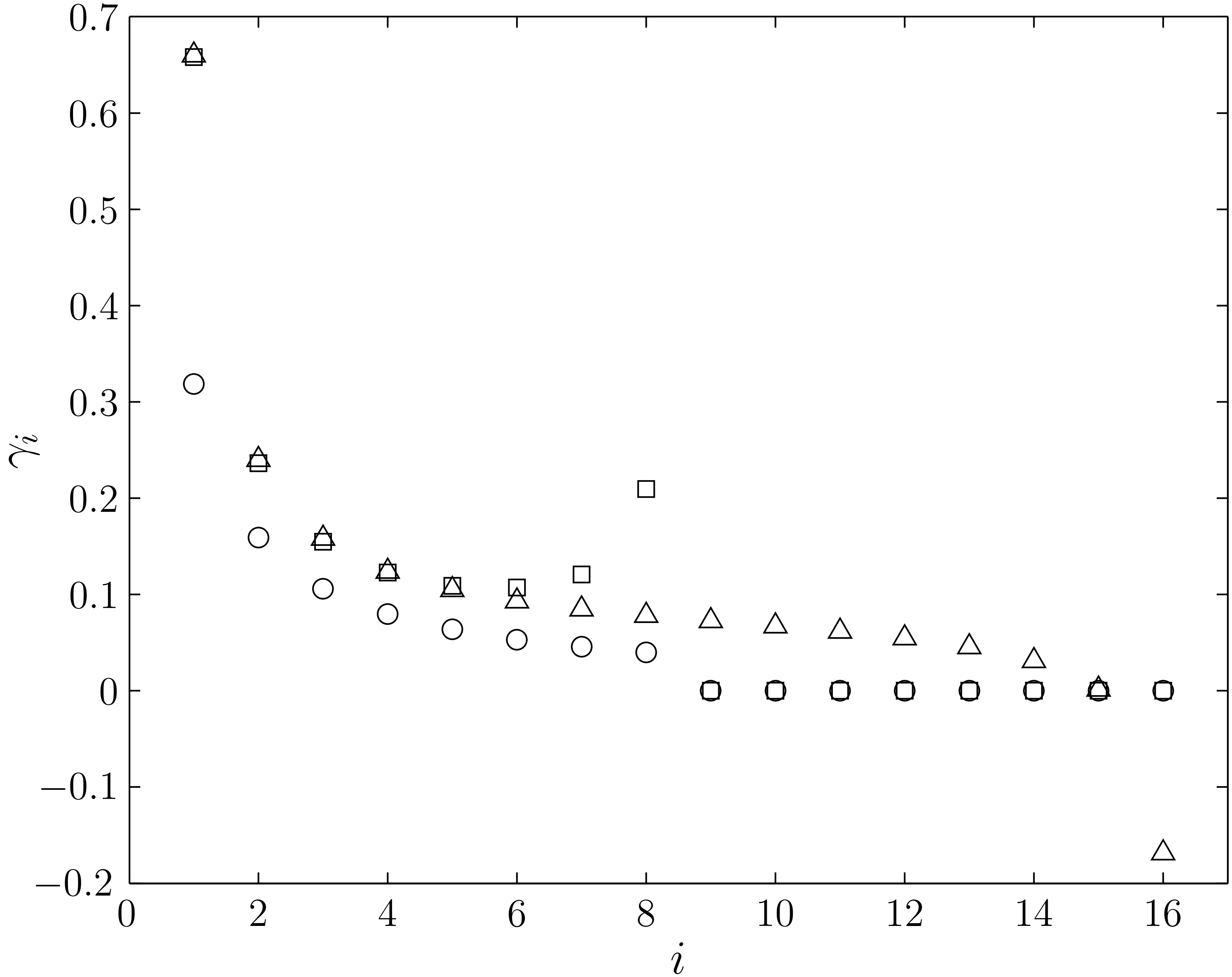
Optimal matrix

Recall we wish to show that $\alpha^+(n) = \sum_{i=0}^{n-1} \left(\frac{\binom{2i}{i}}{4^i} \right)^2$.

Define $A_n(j) = \sum_{i=0}^{n-j-1} \beta_i \beta_{i+j}$.

$$\begin{pmatrix} A_4(0) - A_4(1) & A_4(1) - A_4(2) & A_4(2) - A_4(3) & A_4(3) \\ A_4(1) - A_4(2) & A_4(2) - A_4(3) & A_4(3) & 0 \\ A_4(2) - A_4(3) & A_4(3) & 0 & 0 \\ A_4(3) & 0 & 0 & 0 \end{pmatrix}$$

To bound spectral norm, show that $x = [\beta_3, \beta_2, \beta_1, \beta_0]$ is eigenvector with eigenvalue 1.



0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
γ_i
0
2
4
6
8
10
12
14
16
i

Conclusion

- What is the quantum query complexity of ordered search?
- Progress will require new algorithms or new lower bound techniques.
- [BSS03] show quantum query complexity can be written as a semidefinite program. Adversary bound can be viewed as a relaxation of this program.
- Our optimal matrix can be used to give nearly elementary proof of Hilbert's Inequality (need $\Gamma(1/2) = \sqrt{\pi}$).