Attacking some open questions in quantum query complexity

Ronald de Wolf

and University of Amsterdam

Quantum query algorithms

Quantum query algorithms

- Goal: compute n-bit function f

Quantum query algorithms

- Goal: compute n-bit function f
- T-query quantum algorithm interleaves fixed unitaries with queries to its input $x \in \{0,1\}^n$

Quantum query algorithms

- Goal: compute n-bit function f
- T-query quantum algorithm interleaves fixed unitaries with queries to its input $x \in \{0,1\}^n$

$$O_x : |i, b\rangle \mapsto |i, b \oplus x_i\rangle$$

Quantum query algorithms

- Goal: compute n-bit function f
- T-query quantum algorithm interleaves fixed unitaries with queries to its input $x \in \{0,1\}^n$

$$O_x : |i, b\rangle \mapsto |i, b \oplus x_i\rangle$$

- A final measurement determines output

Quantum query algorithms

- Goal: compute n-bit function f
- T-query quantum algorithm interleaves fixed unitaries with queries to its input $x \in \{0,1\}^n$

$$O_x : |i, b\rangle \mapsto |i, b \oplus x_i\rangle$$

- A final measurement determines output

- Most known quantum algorithms function in this setting:

Quantum query algorithms

- Goal: compute n-bit function f
- T-query quantum algorithm interleaves fixed unitaries with queries to its input $x \in \{0,1\}^n$

$$O_x : |i, b\rangle \mapsto |i, b \oplus x_i\rangle$$

- A final measurement determines output

- Most known quantum algorithms function in this setting:

  Deutsch-Jozsa, Simon, Shor, Grover, random walks

What do we know?

What do we know?

- Exponential gaps between classical and quantum query complexity for some partial functions (Simon, Shor, . . .)

What do we know?

- Exponential gaps between classical and quantum query complexity for some partial functions (Simon, Shor, . . .)
- Quadratic gap for total functions (Grover)

What do we know?

- Exponential gaps between classical and quantum query complexity for some partial functions (Simon, Shor, . . .)
- Quadratic gap for total functions (Grover)
- Polynomial relation between deterministic classical complexity and quantum bounded-error complexity for all total functions: $D(f) = O(Q_2(f)^6)$

What do we know?

- Exponential gaps between classical and quantum query complexity for some partial functions (Simon, Shor, . . .)
- Quadratic gap for total functions (Grover)
- Polynomial relation between deterministic classical complexity and quantum bounded-error complexity for all total functions: $D(f) = O(Q_2(f)^6)$
- Two strong lower bound methods

What do we know?

- Exponential gaps between classical and quantum query complexity for some partial functions (Simon, Shor, . . .)
- Quadratic gap for total functions (Grover)
- Polynomial relation between deterministic classical complexity and quantum bounded-error complexity for all total functions: $D(f) = O(Q_2(f)^6)$
- Two strong lower bound methods
  1. Polynomials [BBCMW'98]

What do we know?

- Exponential gaps between classical and quantum query complexity for some partial functions (Simon, Shor, . . .)
- Quadratic gap for total functions (Grover)
- Polynomial relation between deterministic classical complexity and quantum bounded-error complexity for all total functions: $D(f) = O(Q_2(f)^6)$
- Two strong lower bound methods
  1. Polynomials [BBCMW'98]
  2. Adversary [Ambainis'00]

What do we know?

- Exponential gaps between classical and quantum query complexity for some partial functions (Simon, Shor, . . .)
- Quadratic gap for total functions (Grover)
- Polynomial relation between deterministic classical complexity and quantum bounded-error complexity for all total functions: $D(f) = O(Q_2(f)^6)$
- Two strong lower bound methods
  1. Polynomials [BBCMW'98]
  2. Adversary [Ambainis'00], recently a stronger variant characterized $Q_2(f)$ [Reichardt'09]

What do we know?

- Exponential gaps between classical and quantum query complexity for some partial functions (Simon, Shor, . . .)
- Quadratic gap for total functions (Grover)
- Polynomial relation between deterministic classical complexity and quantum bounded-error complexity for all total functions: $D(f) = O(Q_2(f)^6)$
- Two strong lower bound methods
  1. Polynomials [BBCMW'98]
  2. Adversary [Ambainis'00], recently a stronger variant characterized $Q_2(f)$ [Reichardt'09]
- Algorithms match lower bounds for most problems

What do we know?

- Exponential gaps between classical and quantum query complexity for some partial functions (Simon, Shor, . . .)
- Quadratic gap for total functions (Grover)
- Polynomial relation between deterministic classical complexity and quantum bounded-error complexity for all total functions: $D(f) = O(Q_2(f)^6)$
- Two strong lower bound methods
  1. Polynomials [BBCMW'98]
  2. Adversary [Ambainis'00], recently a stronger variant characterized $Q_2(f)$ [Reichardt'09]
- Algorithms match lower bounds for most problems (except: testing matrices $AB = C$, triangle-finding)

Some open problems

Some open problems

1. Can we improve 6th-power relation between quantum and classical query complexity?

Some open problems

1. Can we improve 6th-power relation between quantum and classical query complexity?

2. Do all total f that depend on all n bits need $\Omega(\log n)$ queries?

Some open problems

1. Can we improve 6th-power relation between quantum and classical query complexity?

2. Do all total f that depend on all n bits need $\Omega(\log n)$ queries?

3. Do almost all total f need $\geq n/2$ queries?

Some open problems

1. Can we improve 6th-power relation between quantum and classical query complexity?

2. Do all total f that depend on all n bits need $\Omega(\log n)$ queries?

3. Do almost all total f need $\geq n/2$ queries?

   Partial answer: yes if the algorithm uses small space

Polynomial relation $D(f)$ vs $Q_2(f)$

Polynomial relation $D(f)$ vs $Q_2(f)$

- Classical deterministic and quantum bounded-error query complexity are polynomially related for all total f

$$D(f) = O(Q_2(f)^6)$$

Polynomial relation $D(f)$ vs $Q_2(f)$

- Classical deterministic and quantum bounded-error query complexity are polynomially related for all total f

$$D(f) = O(Q_2(f)^6)$$

- Optimal result is probably $O(Q_2(f)^2)$

Polynomial relation $D(f)$ vs $Q_2(f)$

- Classical deterministic and quantum bounded-error query complexity are polynomially related for all total f

$$D(f) = O(Q_2(f)^6)$$

- Optimal result is probably $O(Q_2(f)^2)$
- This would be tight because of Grover

Polynomial relation $D(f)$ vs $Q_2(f)$

- Classical deterministic and quantum bounded-error query complexity are polynomially related for all total f

$$D(f) = O(Q_2(f)^6)$$

- Optimal result is probably $O(Q_2(f)^2)$
- This would be tight because of Grover
- At least improving it to $Q_2(f)^4$ should be doable

Polynomial relation $D(f)$ vs $Q_2(f)$

- Classical deterministic and quantum bounded-error query complexity are polynomially related for all total f

$$D(f) = O(Q_2(f)^6)$$

- Optimal result is probably $O(Q_2(f)^2)$
- This would be tight because of Grover
- At least improving it to $Q_2(f)^4$ should be doable
- How?

Polynomial relation $D(f)$ vs $Q_2(f)$

- Classical deterministic and quantum bounded-error query complexity are polynomially related for all total f

$$D(f) = O(Q_2(f)^6)$$

- Optimal result is probably $O(Q_2(f)^2)$
- This would be tight because of Grover
- At least improving it to $Q_2(f)^4$ should be doable
- How? We know $D(f) \leq C(f)^2$.

Polynomial relation $D(f)$ vs $Q_2(f)$

- Classical deterministic and quantum bounded-error query complexity are polynomially related for all total f

$$D(f) = O(Q_2(f)^6)$$

- Optimal result is probably $O(Q_2(f)^2)$
- This would be tight because of Grover
- At least improving it to $Q_2(f)^4$ should be doable
- How? We know $D(f) \leq C(f)^2$.
  Use new adversary tools to show $\sqrt{C(f)} \leq Q_2(f)$?

Quantum lower bounds by polynomials

Quantum lower bounds by polynomials

- Amplitudes of final state of a T-query algorithm are n-variate polynomials $\alpha_z(x)$ of degree $\leq T$

Quantum lower bounds by polynomials

- Amplitudes of final state of a T-query algorithm are n-variate polynomials $\alpha_z(x)$ of degree $\leq T$
- Suppose algorithm measures and outputs the first qubit of the final state

Quantum lower bounds by polynomials

- Amplitudes of final state of a T-query algorithm are n-variate polynomials $\alpha_z(x)$ of degree $\leq T$
- Suppose algorithm measures and outputs the first qubit of the final state: $\Pr[\text{algo outputs } 1] = \sum_{z:z_1=1} |\alpha_z(x)|^2$

Quantum lower bounds by polynomials

- Amplitudes of final state of a T-query algorithm are n-variate polynomials $\alpha_z(x)$ of degree $\leq T$
- Suppose algorithm measures and outputs the first qubit of the final state: $\Pr[\text{algo outputs } 1] = \sum_{z:z_1=1} |\alpha_z(x)|^2$

  This is a polynomial $P(x)$ of degree $\leq 2T$

Quantum lower bounds by polynomials

- Amplitudes of final state of a T-query algorithm are n-variate polynomials $\alpha_z(x)$ of degree $\leq T$
- Suppose algorithm measures and outputs the first qubit of the final state: $\Pr[\text{algo outputs } 1] = \sum_{z:z_1=1} |\alpha_z(x)|^2$

  This is a polynomial $P(x)$ of degree $\leq 2T$
- If algorithm computes $f : \{0,1\}^n \to \{0,1\}$ with error ε, then $|P(x) - f(x)| \leq \varepsilon$ for all $x \in \{0,1\}^n$

Quantum lower bounds by polynomials

- Amplitudes of final state of a T-query algorithm are n-variate polynomials $\alpha_z(x)$ of degree $\leq T$
- Suppose algorithm measures and outputs the first qubit of the final state: $\Pr[\text{algo outputs } 1] = \sum_{z:z_1=1} |\alpha_z(x)|^2$

  This is a polynomial $P(x)$ of degree $\leq 2T$
- If algorithm computes $f : \{0,1\}^n \to \{0,1\}$ with error ε, then $|P(x) - f(x)| \leq \varepsilon$ for all $x \in \{0,1\}^n$
- Degree lower bounds imply query lower bounds:

Quantum lower bounds by polynomials

- Amplitudes of final state of a T-query algorithm are n-variate polynomials $\alpha_z(x)$ of degree $\leq T$
- Suppose algorithm measures and outputs the first qubit of the final state: $\Pr[\text{algo outputs } 1] = \sum_{z:z_1=1} |\alpha_z(x)|^2$

  This is a polynomial $P(x)$ of degree $\leq 2T$
- If algorithm computes $f : \{0,1\}^n \to \{0,1\}$ with error ε, then $|P(x) - f(x)| \leq \varepsilon$ for all $x \in \{0,1\}^n$
- Degree lower bounds imply query lower bounds:

  $Q_E(f) \geq \frac{1}{2} deg(f)$ for $\varepsilon = 0$

Quantum lower bounds by polynomials

- Amplitudes of final state of a T-query algorithm are n-variate polynomials $\alpha_z(x)$ of degree $\leq T$
- Suppose algorithm measures and outputs the first qubit of the final state: $\Pr[\text{algo outputs } 1] = \sum_{z:z_1=1} |\alpha_z(x)|^2$

  This is a polynomial $P(x)$ of degree $\leq 2T$
- If algorithm computes $f : \{0,1\}^n \to \{0,1\}$ with error ε, then $|P(x) - f(x)| \leq \varepsilon$ for all $x \in \{0,1\}^n$
- Degree lower bounds imply query lower bounds:

  $Q_E(f) \geq \frac{1}{2} deg(f)$ for $\varepsilon = 0$

  $Q_2(f) \geq \frac{1}{2} \widetilde{deg}(f)$ for $\varepsilon = 1/3$

Lower bounds for all functions

Lower bounds for all functions

- Suppose $f : \{0,1\}^n \to \{0,1\}$ depends on all n inputs bits

Lower bounds for all functions

- Suppose $f : \{0,1\}^n \to \{0,1\}$ depends on all n inputs bits
- Influence of i-th input bit: $\mathrm{Inf}_i(f) := \Pr_x[f(x) \neq f(x \oplus e_i)]$

Lower bounds for all functions

- Suppose $f : \{0,1\}^n \to \{0,1\}$ depends on all n inputs bits
- Influence of i-th input bit: $\mathrm{Inf}_i(f) := \Pr_x[f(x) \neq f(x \oplus e_i)]$
- Schwartz-Zippel lemma: $\mathrm{Inf}_i(f) \geq 2^{-deg(f)}$ for all i

Lower bounds for all functions

- Suppose $f : \{0,1\}^n \to \{0,1\}$ depends on all n inputs bits
- Influence of i-th input bit: $\mathrm{Inf}_i(f) := \mathrm{Pr}_x[f(x) \neq f(x \oplus e_i)]$
- Schwartz-Zippel lemma: $\mathrm{Inf}_i(f) \geq 2^{-deg(f)}$ for all i
- $\dfrac{n}{2^{deg(f)}}$

Lower bounds for all functions

- Suppose $f : \{0,1\}^n \to \{0,1\}$ depends on all n inputs bits
- Influence of i-th input bit: $\mathrm{Inf}_i(f) := \Pr_x[f(x) \neq f(x \oplus e_i)]$
- Schwartz-Zippel lemma: $\mathrm{Inf}_i(f) \geq 2^{-deg(f)}$ for all i
- $$\frac{n}{2^{deg(f)}} \leq \sum_{i=1}^{n} \mathrm{Inf}_i(f)$$

Lower bounds for all functions

- Suppose $f : \{0,1\}^n \to \{0,1\}$ depends on all n inputs bits
- Influence of i-th input bit: $\mathrm{Inf}_i(f) := \Pr_x[f(x) \neq f(x \oplus e_i)]$
- Schwartz-Zippel lemma: $\mathrm{Inf}_i(f) \geq 2^{-deg(f)}$ for all i
- $$\frac{n}{2^{deg(f)}} \leq \sum_{i=1}^{n} \mathrm{Inf}_i(f) \leq deg(f)$$

Lower bounds for all functions

- Suppose $f : \{0,1\}^n \to \{0,1\}$ depends on all n inputs bits
- Influence of i-th input bit: $\mathrm{Inf}_i(f) := \Pr_x[f(x) \neq f(x \oplus e_i)]$
- Schwartz-Zippel lemma: $\mathrm{Inf}_i(f) \geq 2^{-deg(f)}$ for all i
- $\frac{n}{2^{deg(f)}} \leq \sum_{i=1}^{n} \mathrm{Inf}_i(f) \leq deg(f)$

  $\Rightarrow deg(f) \geq \log n - O(\log \log n)$ [Nisan-Szegedy'92]

Lower bounds for all functions

- Suppose $f : \{0,1\}^n \to \{0,1\}$ depends on all n inputs bits
- Influence of i-th input bit: $\mathrm{Inf}_i(f) := \Pr_x[f(x) \neq f(x \oplus e_i)]$
- Schwartz-Zippel lemma: $\mathrm{Inf}_i(f) \geq 2^{-deg(f)}$ for all i
- $$\frac{n}{2^{deg(f)}} \leq \sum_{i=1}^{n} \mathrm{Inf}_i(f) \leq deg(f)$$
  $\Rightarrow deg(f) \geq \log n - O(\log \log n)$ [Nisan-Szegedy'92]
  $\Rightarrow Q_E(f) \geq \frac{1}{2} \log n - O(\log \log n)$

Lower bounds for all functions

- Suppose $f : \{0,1\}^n \to \{0,1\}$ depends on all n inputs bits
- Influence of i-th input bit: $\mathrm{Inf}_i(f) := \Pr_x[f(x) \neq f(x \oplus e_i)]$
- Schwartz-Zippel lemma: $\mathrm{Inf}_i(f) \geq 2^{-deg(f)}$ for all i
- $\frac{n}{2^{deg(f)}} \leq \sum_{i=1}^{n} \mathrm{Inf}_i(f) \leq deg(f)$

  $\Rightarrow deg(f) \geq \log n - O(\log \log n)$ [Nisan-Szegedy'92]

  $\Rightarrow Q_E(f) \geq \frac{1}{2} \log n - O(\log \log n)$
- What about $Q_2(f)$ and $\widetilde{deg}(f)$?

Lower bounds for all functions

- Suppose $f : \{0,1\}^n \to \{0,1\}$ depends on all n inputs bits
- Influence of i-th input bit: $\mathrm{Inf}_i(f) := \Pr_x[f(x) \neq f(x \oplus e_i)]$
- Schwartz-Zippel lemma: $\mathrm{Inf}_i(f) \geq 2^{-deg(f)}$ for all i
- $\frac{n}{2^{deg(f)}} \leq \sum_{i=1}^{n} \mathrm{Inf}_i(f) \leq deg(f)$

  $\Rightarrow deg(f) \geq \log n - O(\log \log n)$ [Nisan-Szegedy'92]

  $\Rightarrow Q_E(f) \geq \frac{1}{2} \log n - O(\log \log n)$
- What about $Q_2(f)$ and $\widetilde{deg}(f)$?

  $\sum_{i=1}^{n} \mathrm{Inf}_i(f) \leq \widetilde{deg}(f)$ still works;

Lower bounds for all functions

- Suppose $f : \{0,1\}^n \to \{0,1\}$ depends on all n inputs bits
- Influence of i-th input bit: $\mathrm{Inf}_i(f) := \Pr_x[f(x) \neq f(x \oplus e_i)]$
- Schwartz-Zippel lemma: $\mathrm{Inf}_i(f) \geq 2^{-deg(f)}$ for all i
- $\frac{n}{2^{deg(f)}} \leq \sum_{i=1}^{n} \mathrm{Inf}_i(f) \leq deg(f)$

  $\Rightarrow deg(f) \geq \log n - O(\log \log n)$ [Nisan-Szegedy'92]

  $\Rightarrow Q_E(f) \geq \frac{1}{2} \log n - O(\log \log n)$
- What about $Q_2(f)$ and $\widetilde{deg}(f)$?

  $\sum_{i=1}^{n} \mathrm{Inf}_i(f) \leq \widetilde{deg}(f)$ still works; $\mathrm{Inf}_i(f) \geq 2^{-\widetilde{deg}(f)}$ fails

Lower bounds for all functions

- Suppose $f : \{0,1\}^n \to \{0,1\}$ depends on all n inputs bits
- Influence of i-th input bit: $\mathrm{Inf}_i(f) := \Pr_x[f(x) \neq f(x \oplus e_i)]$
- Schwartz-Zippel lemma: $\mathrm{Inf}_i(f) \geq 2^{-deg(f)}$ for all i
- $\frac{n}{2^{deg(f)}} \leq \sum_{i=1}^{n} \mathrm{Inf}_i(f) \leq deg(f)$

  $\Rightarrow deg(f) \geq \log n - O(\log \log n)$ [Nisan-Szegedy'92]

  $\Rightarrow Q_E(f) \geq \frac{1}{2} \log n - O(\log \log n)$
- What about $Q_2(f)$ and $\widetilde{deg}(f)$?

  $\sum_{i=1}^{n} \mathrm{Inf}_i(f) \leq \widetilde{deg}(f)$ still works; $\mathrm{Inf}_i(f) \geq 2^{-\widetilde{deg}(f)}$ fails
- Gives only $\sqrt{\log n}$ lower bound on $\widetilde{deg}(f)$ and $Q_2(f)$

Extending result to bounded-error?

Extending result to bounded-error?

- Talagrand'94: $\sum_{i=1}^{n} \frac{\mathsf{Inf}_i(f)}{\log(1/\mathsf{Inf}_i(f))} \geq \mathsf{Var}[f]$

Extending result to bounded-error?

- Talagrand'94: $\sum_{i=1}^{n} \frac{\mathsf{Inf}_i(f)}{\log(1/\mathsf{Inf}_i(f))} \geq \mathsf{Var}[f]$
- Aside: a function f with constant variance has at least one variable with $\mathsf{Inf}_i(f) \geq \Omega(\frac{\log n}{n})$ (KKL)

Extending result to bounded-error?

- Talagrand'94: $\sum_{i=1}^{n} \frac{\mathsf{Inf}_i(f)}{\log(1/\mathsf{Inf}_i(f))} \geq \mathsf{Var}[f]$
- Aside: a function f with constant variance has at least one variable with $\mathsf{Inf}_i(f) \geq \Omega(\frac{\log n}{n})$ (KKL)
- If $\mathsf{Var}[f]$ is constant and all $\mathsf{Inf}_i(f) \leq 1/n^{\varepsilon}$

Extending result to bounded-error?

- Talagrand'94: $\sum_{i=1}^{n} \frac{\mathsf{Inf}_i(f)}{\log(1/\mathsf{Inf}_i(f))} \geq \mathsf{Var}[f]$
- Aside: a function f with constant variance has at least one variable with $\mathsf{Inf}_i(f) \geq \Omega(\frac{\log n}{n})$ (KKL)
- If $\mathsf{Var}[f]$ is constant and all $\mathsf{Inf}_i(f) \leq 1/n^{\varepsilon}$ then Talagrand implies $\sum_{i=1}^{n} \mathsf{Inf}_i(f) = \Omega(\log n)$ and we're done

Extending result to bounded-error?

- Talagrand'94: $\sum_{i=1}^{n} \frac{\mathsf{Inf}_i(f)}{\log(1/\mathsf{Inf}_i(f))} \geq \mathsf{Var}[f]$
- Aside: a function f with constant variance has at least one variable with $\mathsf{Inf}_i(f) \geq \Omega(\frac{\log n}{n})$ (KKL)
- If $\mathsf{Var}[f]$ is constant and all $\mathsf{Inf}_i(f) \leq 1/n^{\varepsilon}$ then Talagrand implies $\sum_{i=1}^{n} \mathsf{Inf}_i(f) = \Omega(\log n)$ and we're done
- Idea 1: make $\mathsf{Var}[f]$ constant by combining f with an AND or OR function on a few variables

Extending result to bounded-error?

- Talagrand'94: $\sum_{i=1}^{n} \frac{\mathrm{Inf}_i(f)}{\log(1/\mathrm{Inf}_i(f))} \geq \mathrm{Var}[f]$
- Aside: a function f with constant variance has at least one variable with $\mathrm{Inf}_i(f) \geq \Omega(\frac{\log n}{n})$ (KKL)
- If $\mathrm{Var}[f]$ is constant and all $\mathrm{Inf}_i(f) \leq 1/n^{\varepsilon}$ then Talagrand implies $\sum_{i=1}^{n} \mathrm{Inf}_i(f) = \Omega(\log n)$ and we're done
- Idea 1: make $\mathrm{Var}[f]$ constant by combining f with an AND or OR function on a few variables
- Idea 2: if $\sum_{i=1}^{n} \mathrm{Inf}_i(f) \ll \log n$ then there are only few high-influence variables; fix those in some way

Extending result to bounded-error?

- Talagrand'94: $\sum_{i=1}^{n} \frac{\mathrm{Inf}_i(f)}{\log(1/\mathrm{Inf}_i(f))} \geq \mathrm{Var}[f]$
- Aside: a function f with constant variance has at least one variable with $\mathrm{Inf}_i(f) \geq \Omega(\frac{\log n}{n})$ (KKL)
- If $\mathrm{Var}[f]$ is constant and all $\mathrm{Inf}_i(f) \leq 1/n^{\varepsilon}$ then Talagrand implies $\sum_{i=1}^{n} \mathrm{Inf}_i(f) = \Omega(\log n)$ and we're done
- Idea 1: make $\mathrm{Var}[f]$ constant by combining f with an AND or OR function on a few variables
- Idea 2: if $\sum_{i=1}^{n} \mathrm{Inf}_i(f) \ll \log n$ then there are only few high-influence variables; fix those in some way
- Not so easy to get this to work...

All the information for half the price

All the information for half the price

- van Dam'98: every n-bit function can be computed with $\sim n/2$ queries

All the information for half the price

- van Dam'98: every n-bit function can be computed with $\sim n/2$ queries, because we can recover x whp

All the information for half the price

- van Dam'98: every n-bit function can be computed with $\sim n/2$ queries, because we can recover x whp
- If we had $|\phi\rangle = \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle$

All the information for half the price

- van Dam'98: every n-bit function can be computed with $\sim n/2$ queries, because we can recover x whp
- If we had $|\phi\rangle = \sum_{y\in\{0,1\}^n}(-1)^{x\cdot y}|y\rangle$,
  then Hadamard would give x with probability 1

All the information for half the price

- van Dam'98: every n-bit function can be computed with $\sim n/2$ queries, because we can recover x whp
- If we had $|\phi\rangle = \sum_{y\in\{0,1\}^n}(-1)^{x\cdot y}|y\rangle$,
  then Hadamard would give x with probability 1
- But preparing $|\phi\rangle$ takes n queries

All the information for half the price

- van Dam'98: every n-bit function can be computed with $\sim n/2$ queries, because we can recover x whp
- If we had $|\phi\rangle = \sum_{y\in\{0,1\}^n}(-1)^{x\cdot y}|y\rangle$, then Hadamard would give x with probability 1
- But preparing $|\phi\rangle$ takes n queries
- Trick: instead prepare $|\phi'\rangle = \sum_{y:|y|\leq n/2+2\sqrt{n}}(-1)^{x\cdot y}|y\rangle$

All the information for half the price

- van Dam'98: every n-bit function can be computed with $\sim n/2$ queries, because we can recover x whp
- If we had $|\phi\rangle = \sum_{y\in\{0,1\}^n}(-1)^{x\cdot y}|y\rangle$,
  then Hadamard would give x with probability 1
- But preparing $|\phi\rangle$ takes n queries
- Trick: instead prepare $|\phi'\rangle = \sum_{y:|y|\leq n/2+2\sqrt{n}}(-1)^{x\cdot y}|y\rangle$
- This can be done with $n/2 + 2\sqrt{n}$ queries:

All the information for half the price

- van Dam'98: every n-bit function can be computed with $\sim n/2$ queries, because we can recover x whp
- If we had $|\phi\rangle = \sum_{y\in\{0,1\}^n}(-1)^{x\cdot y}|y\rangle$,
  then Hadamard would give x with probability 1
- But preparing $|\phi\rangle$ takes n queries
- Trick: instead prepare $|\phi'\rangle = \sum_{y:|y|\leq n/2+2\sqrt{n}}(-1)^{x\cdot y}|y\rangle$
- This can be done with $n/2+2\sqrt{n}$ queries:
  (1) prepare $\sum_{y:|y|\leq n/2+2\sqrt{n}}|y\rangle$

All the information for half the price

- van Dam'98: every n-bit function can be computed with $\sim n/2$ queries, because we can recover x whp
- If we had $|\phi\rangle = \sum_{y\in\{0,1\}^n}(-1)^{x\cdot y}|y\rangle$,
  then Hadamard would give x with probability 1
- But preparing $|\phi\rangle$ takes n queries
- Trick: instead prepare $|\phi'\rangle = \sum_{y:|y|\leq n/2+2\sqrt{n}}(-1)^{x\cdot y}|y\rangle$
- This can be done with $n/2+2\sqrt{n}$ queries:
  (1) prepare $\sum_{y:|y|\leq n/2+2\sqrt{n}}|y\rangle$
  (2) map $|y\rangle \mapsto (-1)^{x\cdot y}|y\rangle$ using $n/2+2\sqrt{n}$ queries

All the information for half the price

- van Dam'98: every n-bit function can be computed with $\sim n/2$ queries, because we can recover x whp
- If we had $|\phi\rangle = \sum_{y\in\{0,1\}^n}(-1)^{x\cdot y}|y\rangle$,
  then Hadamard would give x with probability 1
- But preparing $|\phi\rangle$ takes n queries
- Trick: instead prepare $|\phi'\rangle = \sum_{y:|y|\leq n/2+2\sqrt{n}}(-1)^{x\cdot y}|y\rangle$
- This can be done with $n/2+2\sqrt{n}$ queries:
  (1) prepare $\sum_{y:|y|\leq n/2+2\sqrt{n}}|y\rangle$
  (2) map $|y\rangle \mapsto (-1)^{x\cdot y}|y\rangle$ using $n/2+2\sqrt{n}$ queries
- $\langle\phi|\phi'\rangle \geq 0.99$

All the information for half the price

- van Dam'98: every n-bit function can be computed with $\sim n/2$ queries, because we can recover x whp
- If we had $|\phi\rangle = \sum_{y\in\{0,1\}^n}(-1)^{x\cdot y}|y\rangle$,
  then Hadamard would give x with probability 1
- But preparing $|\phi\rangle$ takes n queries
- Trick: instead prepare $|\phi'\rangle = \sum_{y:|y|\leq n/2+2\sqrt{n}}(-1)^{x\cdot y}|y\rangle$
- This can be done with $n/2+2\sqrt{n}$ queries:
  (1) prepare $\sum_{y:|y|\leq n/2+2\sqrt{n}}|y\rangle$
  (2) map $|y\rangle \mapsto (-1)^{x\cdot y}|y\rangle$ using $n/2+2\sqrt{n}$ queries
- $\langle\phi|\phi'\rangle \geq 0.99$, so Hadamard on $|\phi'\rangle$ gives x whp

All the information for half the price

- van Dam'98: every n-bit function can be computed with $\sim n/2$ queries, because we can recover x whp
- If we had $|\phi\rangle = \sum_{y\in\{0,1\}^n}(-1)^{x\cdot y}|y\rangle$,
  then Hadamard would give x with probability 1
- But preparing $|\phi\rangle$ takes n queries
- Trick: instead prepare $|\phi'\rangle = \sum_{y:|y|\leq n/2+2\sqrt{n}}(-1)^{x\cdot y}|y\rangle$
- This can be done with $n/2+2\sqrt{n}$ queries:
  (1) prepare $\sum_{y:|y|\leq n/2+2\sqrt{n}}|y\rangle$
  (2) map $|y\rangle \mapsto (-1)^{x\cdot y}|y\rangle$ using $n/2+2\sqrt{n}$ queries
- $\langle\phi|\phi'\rangle \geq 0.99$, so Hadamard on $|\phi'\rangle$ gives x whp
- Conjecture: almost all n-bit f need $\sim n/2$ queries

What do we know?

What do we know?

- Ambainis'99: almost all $f : \{0,1\}^n \to \{0,1\}$ have approximate degree $\widetilde{deg}(f) \geq n/2$

What do we know?

- Ambainis'99: almost all $f : \{0,1\}^n \to \{0,1\}$ have approximate degree $\widetilde{deg}(f) \geq n/2$
- This implies query complexity $Q_2(f) \geq n/4$

What do we know?

- Ambainis'99: almost all $f : \{0,1\}^n \to \{0,1\}$ have approximate degree $\widetilde{deg}(f) \geq n/2$
- This implies query complexity $Q_2(f) \geq n/4$
- Unfortunately the degree bound is tight: $\widetilde{deg}(f) \approx n/2$ for almost all f

What do we know?

- Ambainis'99: almost all $f : \{0,1\}^n \to \{0,1\}$ have approximate degree $\widetilde{deg}(f) \geq n/2$
- This implies query complexity $Q_2(f) \geq n/4$
- Unfortunately the degree bound is tight: $\widetilde{deg}(f) \approx n/2$ for almost all f
- Could $n/4$ queries suffice?

What do we know?

- Ambainis'99: almost all $f : \{0,1\}^n \to \{0,1\}$ have approximate degree $\widetilde{deg}(f) \geq n/2$
- This implies query complexity $Q_2(f) \geq n/4$
- Unfortunately the degree bound is tight: $\widetilde{deg}(f) \approx n/2$ for almost all f
- Could $n/4$ queries suffice?
- That is indeed the case for *unbounded-error* quantum query complexity

What do we know?

- Ambainis'99: almost all $f : \{0,1\}^n \to \{0,1\}$ have approximate degree $\widetilde{deg}(f) \geq n/2$
- This implies query complexity $Q_2(f) \geq n/4$
- Unfortunately the degree bound is tight: $\widetilde{deg}(f) \approx n/2$ for almost all f
- Could $n/4$ queries suffice?
- That is indeed the case for *unbounded-error* quantum query complexity
- But probably not for bounded-error...

Can prove conjecture for small space

Can prove conjecture for small space

- Polynomial method treats acceptance prob $P(x)$ of T-query algorithm as arbitrary degree-$2T$ polynomial

Can prove conjecture for small space

- Polynomial method treats acceptance prob $P(x)$ of T-query algorithm as arbitrary degree-$2T$ polynomial
- To gain the factor of 2, need to use the special property that P is a sum of squares of degree-T polynomials:

$$P(x) = \sum_{z \in \{0,1\}^S : z_1 = 1} |\alpha_z(x)|^2$$

Can prove conjecture for small space

- Polynomial method treats acceptance prob $P(x)$ of T-query algorithm as arbitrary degree-$2T$ polynomial
- To gain the factor of 2, need to use the special property that P is a sum of squares of degree-T polynomials:

$$P(x) = \sum_{z \in \{0,1\}^S : z_1 = 1} |\alpha_z(x)|^2$$

- Fix degree-T polynomial α_z;

Can prove conjecture for small space

- Polynomial method treats acceptance prob $P(x)$ of T-query algorithm as arbitrary degree-$2T$ polynomial
- To gain the factor of 2, need to use the special property that P is a sum of squares of degree-T polynomials:

$$P(x) = \sum_{z \in \{0,1\}^S : z_1 = 1} |\alpha_z(x)|^2$$

- Fix degree-T polynomial α_z; $B := \sum_{i=0}^{T} \binom{n}{i}$ monomials

Can prove conjecture for small space

- Polynomial method treats acceptance prob $P(x)$ of T-query algorithm as arbitrary degree-$2T$ polynomial
- To gain the factor of 2, need to use the special property that P is a sum of squares of degree-T polynomials:

$$P(x) = \sum_{z \in \{0,1\}^S : z_1 = 1} |\alpha_z(x)|^2$$

- Fix degree-T polynomial α_z; $B := \sum_{i=0}^{T} \binom{n}{i}$ monomials
- With sufficient precision for the coefficients, there are roughly B^B distinct such polynomials

Can prove conjecture for small space

- Polynomial method treats acceptance prob $P(x)$ of T-query algorithm as arbitrary degree-$2T$ polynomial
- To gain the factor of 2, need to use the special property that P is a sum of squares of degree-T polynomials:

$$P(x) = \sum_{z \in \{0,1\}^S : z_1 = 1} |\alpha_z(x)|^2$$

- Fix degree-T polynomial α_z; $B := \sum_{i=0}^{T} \binom{n}{i}$ monomials
- With sufficient precision for the coefficients, there are roughly B^B distinct such polynomials
- Next slide: if $T \ll n/2$ and we choose f randomly, then whp none of the $|\alpha_z|^2$ correlates well with f.

Can prove conjecture for small space

- Polynomial method treats acceptance prob $P(x)$ of T-query algorithm as arbitrary degree-$2T$ polynomial
- To gain the factor of 2, need to use the special property that P is a sum of squares of degree-T polynomials:

$$P(x) = \sum_{z \in \{0,1\}^S : z_1 = 1} |\alpha_z(x)|^2$$

- Fix degree-T polynomial α_z; $B := \sum_{i=0}^{T} \binom{n}{i}$ monomials
- With sufficient precision for the coefficients, there are roughly B^B distinct such polynomials
- Next slide: if $T \ll n/2$ and we choose f randomly, then whp none of the $|\alpha_z|^2$ correlates well with f. Then P can't correlate well with f either

Proof (cntd)

Proof (cntd)

- How well can $|\alpha_z|^2$ correlate with random $f : \{0,1\}^n \rightarrow \{\pm 1\}$?

Proof (cntd)

- How well can $|\alpha_z|^2$ correlate with random $f : \{0,1\}^n \rightarrow \{\pm 1\}$? By Hoeffding bound:

$$\Pr_f \left[\sum_{x \in \{0,1\}^n} |\alpha_z(x)|^2 f(x) > \sqrt{B \log(B) 2^n} \right] \ll 1/B^B.$$

Proof (cntd)

- How well can $|\alpha_z|^2$ correlate with random $f : \{0,1\}^n \rightarrow \{\pm 1\}$? By Hoeffding bound:

$$\Pr_f \left[\sum_{x \in \{0,1\}^n} |\alpha_z(x)|^2 f(x) > \sqrt{B \log(B) 2^n} \right] \ll 1/B^B.$$

- Union bound over all B^B possible polynomials α_z: whp over f, $\sum_x |\alpha_z(x)|^2 f(x) \leq \sqrt{B \log(B) 2^n}$ for all z

Proof (cntd)

- How well can $|\alpha_z|^2$ correlate with random $f : \{0,1\}^n \rightarrow \{\pm 1\}$? By Hoeffding bound:

$$\Pr_f \left[\sum_{x \in \{0,1\}^n} |\alpha_z(x)|^2 f(x) > \sqrt{B \log(B) 2^n} \right] \ll 1/B^B.$$

- Union bound over all B^B possible polynomials α_z: whp over f, $\sum_x |\alpha_z(x)|^2 f(x) \leq \sqrt{B \log(B) 2^n}$ for all z
- $(1-2\varepsilon)2^n$

Proof (cntd)

- How well can $|\alpha_z|^2$ correlate with random $f : \{0,1\}^n \rightarrow \{\pm 1\}$? By Hoeffding bound:

$$\Pr_f \left[\sum_{x \in \{0,1\}^n} |\alpha_z(x)|^2 f(x) > \sqrt{B \log(B) 2^n} \right] \ll 1/B^B.$$

- Union bound over all B^B possible polynomials α_z: whp over f, $\sum_x |\alpha_z(x)|^2 f(x) \leq \sqrt{B \log(B) 2^n}$ for all z
- $(1-2\varepsilon)2^n \leq \sum_x (2P(x)-1)f(x)$

Proof (cntd)

- How well can $|\alpha_z|^2$ correlate with random $f : \{0,1\}^n \rightarrow \{\pm 1\}$? By Hoeffding bound:

$$\Pr_f \left[\sum_{x \in \{0,1\}^n} |\alpha_z(x)|^2 f(x) > \sqrt{B \log(B) 2^n} \right] \ll 1/B^B.$$

- Union bound over all B^B possible polynomials α_z: whp over f, $\sum_x |\alpha_z(x)|^2 f(x) \leq \sqrt{B \log(B) 2^n}$ for all z
- $(1-2\varepsilon)2^n \leq \sum_x (2P(x)-1)f(x) = 2\sum_x P(x)f(x) - \sum_x f(x)$

Proof (cntd)

- How well can $|\alpha_z|^2$ correlate with random $f : \{0,1\}^n \rightarrow \{\pm 1\}$? By Hoeffding bound:

$$\Pr_f \left[\sum_{x \in \{0,1\}^n} |\alpha_z(x)|^2 f(x) > \sqrt{B \log(B) 2^n} \right] \ll 1/B^B.$$

- Union bound over all B^B possible polynomials α_z: whp over f, $\sum_x |\alpha_z(x)|^2 f(x) \leq \sqrt{B \log(B) 2^n}$ for all z
- $$(1-2\varepsilon)2^n \leq \sum_x (2P(x)-1)f(x) = 2\sum_x P(x)f(x) - \sum_x f(x)$$
$$= 2\sum_{x,z} |\alpha_z(x)|^2 f(x) - \sum_x f(x)$$

Proof (cntd)

- How well can $|\alpha_z|^2$ correlate with random $f : \{0,1\}^n \rightarrow \{\pm 1\}$? By Hoeffding bound:

$$\Pr_f \left[\sum_{x \in \{0,1\}^n} |\alpha_z(x)|^2 f(x) > \sqrt{B \log(B) 2^n} \right] \ll 1/B^B.$$

- Union bound over all B^B possible polynomials α_z: whp over f, $\sum_x |\alpha_z(x)|^2 f(x) \leq \sqrt{B \log(B) 2^n}$ for all z
- $$(1-2\varepsilon)2^n \leq \sum_x (2P(x)-1)f(x) = 2\sum_x P(x)f(x) - \sum_x f(x)$$
$$= 2\sum_{x,z} |\alpha_z(x)|^2 f(x) - \sum_x f(x) \leq 2^S \sqrt{B \log(B) 2^n}$$

Proof (cntd)

- How well can $|\alpha_z|^2$ correlate with random $f : \{0,1\}^n \rightarrow \{\pm 1\}$? By Hoeffding bound:

$$\Pr_f \left[\sum_{x \in \{0,1\}^n} |\alpha_z(x)|^2 f(x) > \sqrt{B \log(B) 2^n} \right] \ll 1/B^B.$$

- Union bound over all B^B possible polynomials α_z: whp over f, $\sum_x |\alpha_z(x)|^2 f(x) \leq \sqrt{B \log(B) 2^n}$ for all z
- $$(1-2\varepsilon)2^n \leq \sum_x (2P(x)-1)f(x) = 2\sum_x P(x)f(x) - \sum_x f(x)$$
$$= 2\sum_{x,z} |\alpha_z(x)|^2 f(x) - \sum_x f(x) \leq 2^S \sqrt{B \log(B) 2^n}$$
- $B \log(B) 2^{2S} = \Omega(2^n)$

Proof (cntd)

- How well can $|\alpha_z|^2$ correlate with random $f : \{0,1\}^n \rightarrow \{\pm 1\}$? By Hoeffding bound:

$$\Pr_f \left[\sum_{x \in \{0,1\}^n} |\alpha_z(x)|^2 f(x) > \sqrt{B \log(B) 2^n} \right] \ll 1/B^B.$$

- Union bound over all B^B possible polynomials α_z: whp over f, $\sum_x |\alpha_z(x)|^2 f(x) \leq \sqrt{B \log(B) 2^n}$ for all z
- $$(1-2\varepsilon)2^n \leq \sum_x (2P(x)-1)f(x) = 2\sum_x P(x)f(x) - \sum_x f(x)$$
$$= 2\sum_{x,z} |\alpha_z(x)|^2 f(x) - \sum_x f(x) \leq 2^S \sqrt{B \log(B) 2^n}$$
- $B \log(B) 2^{2S} = \Omega(2^n)$, hence $T \geq (\frac{1}{2} - o(1))n$ if $S = o(n)$

Summary

Summary

- Three open problems in quantum query complexity:

Summary

- Three open problems in quantum query complexity:

  1. Improve $D(f) \leq O(Q_2(f)^6)$ to $Q_2(f)^4$, or even $Q_2(f)^2$

Summary

- Three open problems in quantum query complexity:

  1. Improve $D(f) \leq O(Q_2(f)^6)$ to $Q_2(f)^4$, or even $Q_2(f)^2$

  2. Show that all functions that depend on n variables have $Q_2(f) = \Omega(\log n)$

Summary

- Three open problems in quantum query complexity:

  1. Improve $D(f) \leq O(Q_2(f)^6)$ to $Q_2(f)^4$, or even $Q_2(f)^2$

  2. Show that all functions that depend on n variables have $Q_2(f) = \Omega(\log n)$

  3. Show that almost all functions have $Q_2(f) \geq n/2$

Summary

- Three open problems in quantum query complexity:

  1. Improve $D(f) \leq O(Q_2(f)^6)$ to $Q_2(f)^4$, or even $Q_2(f)^2$

  2. Show that all functions that depend on n variables have $Q_2(f) = \Omega(\log n)$

  3. Show that almost all functions have $Q_2(f) \geq n/2$
     That would be tight because of van Dam's result

Summary

- Three open problems in quantum query complexity:

  1. Improve $D(f) \leq O(Q_2(f)^6)$ to $Q_2(f)^4$, or even $Q_2(f)^2$

  2. Show that all functions that depend on n variables have $Q_2(f) = \Omega(\log n)$

  3. Show that almost all functions have $Q_2(f) \geq n/2$
     That would be tight because of van Dam's result
     Can prove it for small-space algorithms

Summary

- Three open problems in quantum query complexity:

  1. Improve $D(f) \leq O(Q_2(f)^6)$ to $Q_2(f)^4$, or even $Q_2(f)^2$

  2. Show that all functions that depend on n variables have $Q_2(f) = \Omega(\log n)$

  3. Show that almost all functions have $Q_2(f) \geq n/2$
     That would be tight because of van Dam's result
     Can prove it for small-space algorithms

- Nice connections with analysis of Boolean functions