Negative weights make adversaries stronger

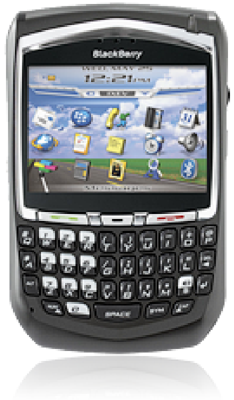
Troy Lee
LRI, Université Paris-Sud

Joint work with: Peter Høyer and Robert Špalek

Quantum query complexity

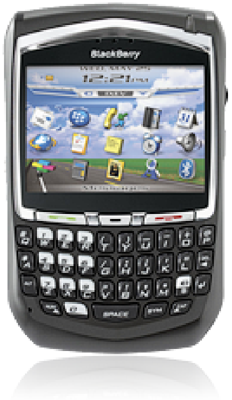
- Popular model for study
- Seems to capture power of quantum computing:
  - Grover's search algorithm,
  - Period finding of Shor's algorithm,
  - Quantum walks: element distinctness, triangle finding, matrix multiplication
- And we can also prove lower bounds!
  - Polynomial method, Quantum Adversary method

Adversary method



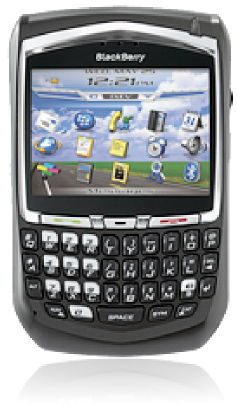
- Adversary method developed by Ambainis, 2002.

Adversary method



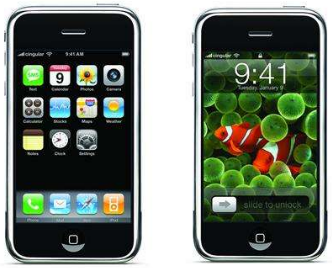
- Adversary method developed by Ambainis, 2002.
- Many competing formulations: weight schemes Amb03, Zha05, spectral norm of matrices BSS03, and Kolmogorov complexity LM04.

Adversary method



- Adversary method developed by Ambainis, 2002.
- Many competing formulations: weight schemes Amb03, Zha05, spectral norm of matrices BSS03, and Kolmogorov complexity LM04.
- All these methods shown equivalent by Špalek and Szegedy, 2006.

Reinventing the adversary



- We introduce a new adversary method, $\mathrm{ADV}^{\pm}$.
- $\mathrm{ADV}^{\pm}(f) \geq \mathrm{ADV}(f)$. We show a function where $\mathrm{ADV}(f) = O(m)$ and $\mathrm{ADV}^{\pm}(f) = \Omega(m^{1.098})$.
- We essentially use that a successful algorithm *computes* a function, not just that it can *distinguish* inputs with different function values.
- Our method does not face the limitations of previous adversary methods

Quantum queries

- In classical query complexity, want to compute $f(x)$ and can make queries of the form $x_i =?$ Complexity is number of queries on worst case input.

- Quantum query—turn query operator into unitary transformation on Hilbert Space $H_I \otimes H_Q \otimes H_W$

$$O|x\rangle|i\rangle|z\rangle \to (-1)^{x_i}|x\rangle|i\rangle|z\rangle.$$

- Can make queries in superposition.

Query algorithm

- On input x, algorithm proceeds by alternating queries and arbitrary unitary transformations independent of x

$$|\phi_x^t\rangle = U_t O U_{t-1} \dots U_1 O U_0 |x\rangle|0\rangle|0\rangle.$$

- Output determined by complete set of orthogonal projectors $\{\Pi_0, \Pi_1\}$. A T-query algorithm outputs b on input x with probability $\|\Pi_b|\phi_x^T\rangle\|^2$.

- $Q_2(f)$ is number T of queries needed by best algorithm which outputs $f(x)$ on input x with probability at least $2/3$, for all x.

Adversary method

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function, and Γ a Hermitian 2^n-by-2^n matrix where $\Gamma[x,y] = 0$ if $f(x) = f(y)$. Then

$$\mathrm{ADV}(f) = \max_{\substack{\Gamma \geq 0 \\ \Gamma \neq 0}} \frac{\|\Gamma\|}{\max_i \|\Gamma \circ D_i\|}.$$

D_i is a zero-one matrix where $D_i[x,y] = 1$ if $x_i \neq y_i$ and $D_i[x,y] = 0$ otherwise.

Theorem [BSS03]: $Q_2(f) = \Omega(\mathrm{ADV}(f))$.

The Γ matrix

	$f^{-1}(0)$	$f^{-1}(1)$
$f^{-1}(0)$	0	A
$f^{-1}(1)$	A^*	0

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

Example: OR function

We define the matrix:

	1000	0100	0010	0001
0000	1	1	1	1

The spectral norm of this matrix is $\sqrt{4}$, and the spectral norm of each $\Gamma \circ D_i$ is one.

Example: OR function

We define the matrix:

	1000	0100	0010	0001
0000	1	1	1	1

The spectral norm of this matrix is $\sqrt{4}$, and the spectral norm of each $\Gamma \circ D_i$ is one.

Generalizing this construction we find $Q_2(\mathrm{OR}_n) = \Omega(\sqrt{n})$.

New adversary method

We remove the restriction to nonnegative matrices:

$$\mathrm{ADV}^{\pm}(f) = \max_{\Gamma \neq 0} \frac{\|\Gamma\|}{\max_i \|\Gamma \circ D_i\|}.$$

Theorem: $Q_2(f) = \Omega(\mathrm{ADV}^{\pm}(f))$.

New adversary method

We remove the restriction to nonnegative matrices:

$$\mathrm{ADV}^{\pm}(f) = \max_{\Gamma \neq 0} \frac{\|\Gamma\|}{\max_i \|\Gamma \circ D_i\|}.$$

Theorem: $Q_2(f) = \Omega(\mathrm{ADV}^{\pm}(f))$.

As we maximize over a larger set, $\mathrm{ADV}^{\pm}(f) \geq \mathrm{ADV}(f)$. It turns out that negative entries can help in giving larger lower bounds!

Automorphism Principle

Let $f : \{0,1\}^n \longrightarrow \{0,1\}$, and let $\pi \in S_n \times \mathbb{Z}_2^n$.

Automorphism Principle

Let $f : \{0,1\}^n \rightarrow \{0,1\}$, and let $\pi \in S_n \times \mathbb{Z}_2^n$. We say that π is an automorphism of f if $f(\pi(x)) = f(x)$ for all $x \in \{0,1\}^n$.

Automorphism Principle

Let $f : \{0,1\}^n \to \{0,1\}$, and let $\pi \in S_n \times \mathbb{Z}_2^n$. We say that π is an automorphism of f if $f(\pi(x)) = f(x)$ for all $x \in \{0,1\}^n$.

- Let G be a group of automorphisms of f. There is an optimal adversary matrix Γ with $\Gamma[x,y] = \Gamma[\pi(x), \pi(y)]$ for all $\pi \in G$.

Automorphism Principle

Let $f : \{0,1\}^n \to \{0,1\}$, and let $\pi \in S_n \times \mathbb{Z}_2^n$. We say that π is an automorphism of f if $f(\pi(x)) = f(x)$ for all $x \in \{0,1\}^n$.

- Let G be a group of automorphisms of f. There is an optimal adversary matrix Γ with $\Gamma[x,y] = \Gamma[\pi(x), \pi(y)]$ for all $\pi \in G$.

- If for all x, y with $f(x) = f(y)$, there exists $\pi \in G$ with $y = \pi(x)$, then the uniform vector is a principal eigenvector of Γ.

Composition principle

Let $f : \{0,1\}^n \to \{0,1\}$. Write $f^1 = f$ and $f^d : \{0,1\}^{n^d} \to \{0,1\}$ be

$$f^d(x) = f(f^{d-1}(x^{(1)}), f^{d-1}(x^{(2)}), \ldots, f^{d-1}(x^{(n)})).$$

Then $\mathrm{ADV}^{\pm}(f^d) \geq \mathrm{ADV}^{\pm}(f)^d$.

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.
- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0,0,0,1)$.
- The zeros: 0000

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.
- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0,0,0,1)$.
- The zeros: 0000

$$0000 \cdot (4321) \times (0,0,0,1)$$

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.

- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0,0,0,1)$.

- The zeros: 0000

$$0000 \cdot (4321) \times (0,0,0,1) = 0001$$

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.

- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0, 0, 0, 1)$.

- The zeros: 0000, 0001

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.
- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0, 0, 0, 1)$.
- The zeros: 0000, 0001

$$0001 \cdot (4321) \times (0, 0, 0, 1)$$

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.

- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0,0,0,1)$.

- The zeros: 0000, 0001

$$0001 \cdot (4321) \times (0,0,0,1) = 0011$$

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.

- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0, 0, 0, 1)$.

- The zeros: 0000, 0001, 0011

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.
- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0, 0, 0, 1)$.
- The zeros: 0000, 0001, 0011

$$0011 \cdot (4321) \times (0, 0, 0, 1)$$

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.
- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0, 0, 0, 1)$.
- The zeros: 0000, 0001, 0011

$$0011 \cdot (4321) \times (0, 0, 0, 1) = 0111$$

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.
- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0,0,0,1)$.
- The zeros: 0000, 0001, 0011, 0111

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.

- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0,0,0,1)$.

- The zeros: 0000, 0001, 0011, 0111

$$0111 \cdot (4321) \times (0,0,0,1)$$

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.
- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0, 0, 0, 1)$.
- The zeros: 0000, 0001, 0011, 0111

$$0111 \cdot (4321) \times (0, 0, 0, 1) = 1111$$

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.
- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0,0,0,1)$.
- The zeros: 0000, 0001, 0011, 0111, 1111

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.

- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0,0,0,1)$.

- The zeros: 0000, 0001, 0011, 0111, 1111

$$1111 \cdot (4321) \times (0,0,0,1)$$

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.
- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0, 0, 0, 1)$.
- The zeros: 0000, 0001, 0011, 0111, 1111

$$1111 \cdot (4321) \times (0, 0, 0, 1) = 1110$$

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.

- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0, 0, 0, 1)$.

- The zeros: 0000, 0001, 0011, 0111, 1111, 1110,

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.
- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0, 0, 0, 1)$.
- The zeros: 0000, 0001, 0011, 0111, 1111, 1110, 1100,

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.
- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0, 0, 0, 1)$.
- The zeros: 0000, 0001, 0011, 0111, 1111, 1110, 1100, 1000.

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.
- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0,0,0,1)$.
- The zeros: 0000, 0001, 0011, 0111, 1111, 1110, 1100, 1000.
- The ones: 0010,

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.
- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0, 0, 0, 1)$.
- The zeros: 0000, 0001, 0011, 0111, 1111, 1110, 1100, 1000.
- The ones: 0010, 0101,

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.
- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0, 0, 0, 1)$.
- The zeros: 0000, 0001, 0011, 0111, 1111, 1110, 1100, 1000.
- The ones: 0010, 0101, 1011,

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.
- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0, 0, 0, 1)$.
- The zeros: 0000, 0001, 0011, 0111, 1111, 1110, 1100, 1000.
- The ones: 0010, 0101, 1011, 0110,

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.
- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0,0,0,1)$.
- The zeros: 0000, 0001, 0011, 0111, 1111, 1110, 1100, 1000.
- The ones: 0010, 0101, 1011, 0110, 1101,

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.
- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0, 0, 0, 1)$.
- The zeros: 0000, 0001, 0011, 0111, 1111, 1110, 1100, 1000.
- The ones: 0010, 0101, 1011, 0110, 1101, 1010,

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.
- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0,0,0,1)$.
- The zeros: 0000, 0001, 0011, 0111, 1111, 1110, 1100, 1000.
- The ones: 0010, 0101, 1011, 0110, 1101, 1010, 0100,

Another example: Ambainis function

- Originally used by Ambainis to separate quantum query complexity from polynomial degree.

- Automorphism group isomorphic to $\mathbb{Z}_8$, generated by $(4321) \times (0, 0, 0, 1)$.

- The zeros: 0000, 0001, 0011, 0111, 1111, 1110, 1100, 1000.

- The ones: 0010, 0101, 1011, 0110, 1101, 1010, 0100, 1001.

Another example: Ambainis function

	0010	0101	1011	0110	1101	1010	0100	1001
0000								
0001								
0011								
0111								
1111								
1110								
1100								
1000								

Another example: Ambainis function

	0010	0101	1011	0110	1101	1010	0100	1001
0000	a							
0001								
0011								
0111								
1111								
1110								
1100								
1000								

Another example: Ambainis function

	0010	0101	1011	0110	1101	1010	0100	1001
0000	a							
0001		a						
0011			a					
0111				a				
1111					a			
1110						a		
1100							a	
1000								a

Another example: Ambainis function

	0010	0101	1011	0110	1101	1010	0100	1001
0000	a	c						
0001		a						
0011			a					
0111				a				
1111					a			
1110						a		
1100							a	
1000								a

Another example: Ambainis function

	0010	0101	1011	0110	1101	1010	0100	1001
0000	a	c						
0001		a	c					
0011			a	c				
0111				a	c			
1111					a	c		
1110						a	c	
1100							a	c
1000	c							a

Another example: Ambainis function

	0010	0101	1011	0110	1101	1010	0100	1001
0000	a	c	d					
0001		a	c	d				
0011			a	c	d			
0111				a	c	d		
1111					a	c	d	
1110						a	c	d
1100	d						a	c
1000	c	d						a

Another example: Ambainis function

	0010	0101	1011	0110	1101	1010	0100	1001
0000	a	c	d	b				
0001		a	c	d	b			
0011			a	c	d	b		
0111				a	c	d	b	
1111					a	c	d	b
1110	b					a	c	d
1100	d	b					a	c
1000	c	d	b					a

Another example: Ambainis function

	0010	0101	1011	0110	1101	1010	0100	1001
0000	a	c	d	b	d			
0001		a	c	d	b	d		
0011			a	c	d	b	d	
0111				a	c	d	b	d
1111	d				a	c	d	b
1110	b	d				a	c	d
1100	d	b	d				a	c
1000	c	d	b	d				a

Another example: Ambainis function

	0010	0101	1011	0110	1101	1010	0100	1001
0000	a	c	d	b	d	c		
0001		a	c	d	b	d	c	
0011			a	c	d	b	d	c
0111	c			a	c	d	b	d
1111	d	c			a	c	d	b
1110	b	d	c			a	c	d
1100	d	b	d	c			a	c
1000	c	d	b	d	c			a

Another example: Ambainis function

	0010	0101	1011	0110	1101	1010	0100	1001
0000	a	c	d	b	d	c	a	
0001		a	c	d	b	d	c	a
0011	a		a	c	d	b	d	c
0111	c	a		a	c	d	b	d
1111	d	c	a		a	c	d	b
1110	b	d	c	a		a	c	d
1100	d	b	d	c	a		a	c
1000	c	d	b	d	c	a		a

Another example: Ambainis function

	0010	0101	1011	0110	1101	1010	0100	1001
0000	a	c	d	b	d	c	a	b
0001	b	a	c	d	b	d	c	a
0011	a	b	a	c	d	b	d	c
0111	c	a	b	a	c	d	b	d
1111	d	c	a	b	a	c	d	b
1110	b	d	c	a	b	a	c	d
1100	d	b	d	c	a	b	a	c
1000	c	d	b	d	c	a	b	a

The $\Gamma \circ D_1$ matrix

	1001	1010	1011	1101
0011	c	b	a	d
0000	b	c	d	d
0001	a	d	c	b
0111	d	d	b	c

Ambainis function continued

We try to maximize $\|\Gamma\| = 2(a + b + c + d)$ while keeping spectral norm of $\Gamma \circ D_i$ at most 1.

	a	b	c	d	$\|\Gamma\|$
ADV	0.75	0.50	0	0	2.5
$\mathrm{ADV}^{\pm}$	0.5788	0.7065	0.1834	-0.2120	2.5136

Ambainis function continued

We try to maximize $\|\Gamma\| = 2(a+b+c+d)$ while keeping spectral norm of $\Gamma \circ D_i$ at most 1.

	a	b	c	d	$\|\Gamma\|$
ADV	0.75	0.50	0	0	2.5
$\mathrm{ADV}^{\pm}$	0.5788	0.7065	0.1834	-0.2120	2.5136

The Ambainis function has polynomial degree 2. By iterating this function, we obtain largest known separation between polynomial degree and quantum query complexity, m vs $m^{1.327}$.

Some words about the proof

Some words about the proof

Recall that running the algorithm on input x for t queries:

$$|\phi_x^t\rangle = U_t O U_{t-1} \dots U_1 O U_0 |x\rangle|0\rangle|0\rangle.$$

Write this as $|\phi_x^t\rangle = |x\rangle|\psi_x^t\rangle$.

Let Γ be an adversary matrix and δ a principal eigenvector.

Some words about the proof

Recall that running the algorithm on input x for t queries:

$$|\phi_x^t\rangle = U_t O U_{t-1} \ldots U_1 O U_0 |x\rangle|0\rangle|0\rangle.$$

Write this as $|\phi_x^t\rangle = |x\rangle|\psi_x^t\rangle$.

Let Γ be an adversary matrix and δ a principal eigenvector. The principal eigenvector *tells us* how to build a hard input— we feed algorithm the superposition $\sum_x \delta_x |x\rangle|0\rangle|0\rangle$.

Some words about the proof

Recall that running the algorithm on input x for t queries:

$$|\phi_x^t\rangle = U_t O U_{t-1} \dots U_1 O U_0 |x\rangle|0\rangle|0\rangle.$$

Write this as $|\phi_x^t\rangle = |x\rangle|\psi_x^t\rangle$.

Let Γ be an adversary matrix and δ a principal eigenvector. The principal eigenvector *tells us* how to build a hard input— we feed algorithm the superposition $\sum_x \delta_x |x\rangle|0\rangle|0\rangle$. State of algorithm after t queries is $\sum_x \delta_x |x\rangle|\psi_x^t\rangle$.

Some words about the proof

Recall that running the algorithm on input x for t queries:

$$|\phi_x^t\rangle = U_t O U_{t-1} \dots U_1 O U_0 |x\rangle|0\rangle|0\rangle.$$

Write this as $|\phi_x^t\rangle = |x\rangle|\psi_x^t\rangle$.

Let Γ be an adversary matrix and δ a principal eigenvector. The principal eigenvector *tells us* how to build a hard input— we feed algorithm the superposition $\sum_x \delta_x |x\rangle|0\rangle|0\rangle$. State of algorithm after t queries is $\sum_x \delta_x |x\rangle|\psi_x^t\rangle$. Let $\rho^{(t)}[x, y] = \delta_x^* \delta_y \langle\psi_x^t|\psi_y^t\rangle$ be the reduced density matrix of this state.

Watch the density matrix. . .

Define a progress function based on $\rho^{(t)}$ as

$$W^{(t)} = \langle \Gamma, \rho^{(t)} \rangle = \sum_{x,y} \Gamma[x,y] \delta_x^* \delta_y \langle \psi_x^t | \psi_y^t \rangle.$$

Watch the density matrix. . .

Define a progress function based on $\rho^{(t)}$ as

$$W^{(t)} = \langle \Gamma, \rho^{(t)} \rangle = \sum_{x,y} \Gamma[x,y] \delta_x^* \delta_y \langle \psi_x^t | \psi_y^t \rangle.$$

Show three things:

- $|W^{(0)}| = \|\Gamma\|$

Watch the density matrix. . .

Define a progress function based on $\rho^{(t)}$ as

$$W^{(t)} = \langle \Gamma, \rho^{(t)} \rangle = \sum_{x,y} \Gamma[x,y] \delta_x^* \delta_y \langle \psi_x^t | \psi_y^t \rangle.$$

Show three things:

- $|W^{(0)}| = \|\Gamma\|$
- $|W^{(T)}| \leq 2\sqrt{\epsilon(1-\epsilon)} \|\Gamma\|$

Watch the density matrix. . .

Define a progress function based on $\rho^{(t)}$ as

$$W^{(t)} = \langle \Gamma, \rho^{(t)} \rangle = \sum_{x,y} \Gamma[x,y] \delta_x^* \delta_y \langle \psi_x^t | \psi_y^t \rangle.$$

Show three things:

- $|W^{(0)}| = \|\Gamma\|$
- $|W^{(T)}| \leq 2\sqrt{\epsilon(1-\epsilon)}\|\Gamma\|$
- $|W^{(t)} - W^{(t+1)}| \leq 2 \max_i \|\Gamma \circ D_i\|$

Normally speaking

- singular values: $\sigma_i(A) = \sqrt{\lambda_i(A^* A)}$
- trace norm: $\|A\|_{tr} = \sum_i \sigma_i(A)$
- Frobenius norm: $\|A\|_F^2 = \sum_i \sigma_i(A)^2 = \sum_{ij} |A[i,j]|^2$
- spectral norm and trace norm are dual:

$$\|A\|_{tr} = \max_B \frac{\langle A, B \rangle}{\|B\|}.$$

Step Two: negative adversary

- Want to upper bound $\langle \Gamma, \rho^{(T)} \rangle \leq 2\sqrt{\epsilon(1-\epsilon)}\|\Gamma\|$.
- Remember $\Gamma \circ F = \Gamma$, where $F[x,y] = 1$ if $f(x) \neq f(y)$ and $F[x,y] = 0$ otherwise.
- Thus $\langle \Gamma, \rho^{(T)} \rangle = \langle \Gamma \circ F, \rho^{(T)} \rangle = \langle \Gamma, \rho^{(T)} \circ F \rangle$.
- Trace norm and spectral norm are dual means: $\langle A, B \rangle \leq \|A\| \cdot \|B\|_{tr}$.

Bounding trace norm of $\rho^{(T)} \circ F$

- Cauchy-Schwarz on singular values: $\|X^*Y\|_{tr} \leq \|X\|_F \|Y\|_F$
- Idea: Factor $\rho^{(T)} \circ F$ into X^*Y, one of which has small Frobenius norm.

Bounding trace norm of $\rho^{(T)} \circ F$

- Cauchy-Schwarz on singular values: $\|X^*Y\|_{tr} \leq \|X\|_F \|Y\|_F$
- Idea: Factor $\rho^{(T)} \circ F$ into X^*Y, one of which has small Frobenius norm.
- Let X be matrix of correct answers: columns of X given be $\delta_x \Pi_{f(x)} |\psi_x^T\rangle$.
  Let Y be matrix of wrong answers: columns of Y given by $\delta_x \Pi_{1-f(x)} |\psi_x^T\rangle$.

Bounding trace norm of $\rho^{(T)} \circ F$

- Cauchy-Schwarz on singular values: $\|X^*Y\|_{tr} \leq \|X\|_F \|Y\|_F$
- Idea: Factor $\rho^{(T)} \circ F$ into X^*Y, one of which has small Frobenius norm.
- Let X be matrix of correct answers: columns of X given be $\delta_x \Pi_{f(x)} |\psi_x^T\rangle$.
  Let Y be matrix of wrong answers: columns of Y given by $\delta_x \Pi_{1-f(x)} |\psi_x^T\rangle$.
- Then $\rho^{(T)} \circ F = X^*Y + Y^*X$.

How that works

- $X^*Y[x,y] = \delta_x^*\delta_y\langle\psi_x^T|\Pi_{f(x)}\Pi_{1-f(y)}|\psi_y^T\rangle$

How that works

- $X^*Y[x,y] = \delta_x^*\delta_y\langle\psi_x^T|\Pi_{f(x)}\Pi_{1-f(y)}|\psi_y^T\rangle$
- $Y^*X[x,y] = \delta_x^*\delta_y\langle\psi_x^T|\Pi_{1-f(x)}\Pi_{f(y)}|\psi_y^T\rangle$

How that works

- $X^*Y[x,y] = \delta_x^*\delta_y\langle\psi_x^T|\Pi_{f(x)}\Pi_{1-f(y)}|\psi_y^T\rangle$
- $Y^*X[x,y] = \delta_x^*\delta_y\langle\psi_x^T|\Pi_{1-f(x)}\Pi_{f(y)}|\psi_y^T\rangle$

$$X^*Y + Y^*X = \begin{cases} \delta_x^*\delta_y\langle\psi_x^T|\psi_y^T\rangle & f(x) \neq f(y) \\ 0 & f(x) = f(y) \end{cases}$$

Bounding trace norm of $\rho^{(T)} \circ F$

- We have $\rho^{(T)} \circ F = X^*Y + Y^*X$ and so $\|\rho^{(T)} \circ F\|_{tr} \leq 2\|X\|_F\|Y\|_F$.
- Norm squared of column x of Y is

$$|\delta_x|^2 \langle \psi_x^T | \Pi_{1-f(x)} | \psi_x^T \rangle \leq \epsilon |\delta_x|^2.$$

Bounding trace norm of $\rho^{(T)} \circ F$

- We have $\rho^{(T)} \circ F = X^*Y + Y^*X$ and so $\|\rho^{(T)} \circ F\|_{tr} \leq 2\|X\|_F\|Y\|_F$.
- Norm squared of column x of Y is

$$|\delta_x|^2 \langle \psi_x^T | \Pi_{1-f(x)} | \psi_x^T \rangle \leq \epsilon |\delta_x|^2.$$

- Thus $\|Y\|_F^2 \leq \epsilon$.
- Further, as $\|X\|_F^2 + \|Y\|_F^2 = 1$, we have $\|\rho^T \circ F\|_{tr} \leq 2\sqrt{\epsilon(1-\epsilon)}$

Open Questions

- Element distinctness: Best bound provable by old method is $\sqrt{2n}$, but right answer is $n^{2/3}$, provable by polynomial method. Can new adversary method prove optimal bound?

- Triangle finding: Best bound provable by old method is n, and best known algorithm gives $n^{1.3}$. Can new adversary bound give a superlinear lower bound?

- $\mathrm{ADV}^{\pm}(f)^2$ is a lower bound on the formula size of f. A challenge is to break the formula size barrier, or show it is not a barrier at all.