EPIT Tutorial
on
Quantum Lower Bounds

Sophie Laplante
LRI, Université Paris-Sud Orsay
I. Overview and model

I Classical, randomized, and quantum query models
II Polynomial method
III Unweighted adversary method
IV Weighted adversary method
V Limits and certificates
VI Polynomial vs adversary
   • Ambainis’ function (adversary > polynomial)
   • Collision (polynomial > adversary)
VII Spectral method, equivalence with weighted method
VIII Lower bounds for formula size.
Adversary method lower bounds

- Spectral method [BSS03]
- Kolmogorov method [LM04]
- Weighted adversary method [A03]
- Adversary method [A02]
- Quantum query complexity
- Semidefinite Programming [BSS03]
- Polynomial method [BBCMW01]

Exact complexity
Strong lower bound
Weak lower bound
To compute a boolean function $f : \{0,1\}^n \to \{0,1\}$,

**Model:** decision tree

**Cost:** Number of queries to input

**Query complexity of $f$:**
depth of shallowest decision tree for $f$
Quantum query model

Query: Unitary transformation $O_x$ that maps $|i, b\rangle$ to $|i, b \oplus x_i\rangle$

Computation:

$$|\psi_T\rangle = U_T O_x \cdots O_x U_0 |0\rangle$$

Output: Measure 1st qubit of $|\psi_T\rangle$

- $Q_E$ (Exact) Error probability at most $1/3$
- $Q_2$ (2-sided error) Error probability at most $1/3$
- For any boolean function $f$, $Q_2(f) \leq Q_E(f)$
Two approaches for lower bounds

**Polynomial method**
- Express the final state on input $x$ as a superposition whose coefficients are multilinear polynomials of degree $T$ in $x$.
- Give a lower bound on the degree of the polynomial that agrees with the function.

**Adversary method**
- Evaluate the difference between the final state on inputs that evaluate to 0 and inputs that evaluate to 1.
- Give a lower bound on how far the states can differ after $T$ steps, in terms of the combinatorial structure of the function.
II. Polynomial Method

- Final state expressed with polynomial coefficients
- Acceptance probability as a polynomial
- 0-error quantum complexity and polynomial degree
- 2-sided error complexity and approximating polynomials
Express the final state on input $x$ as a superposition whose coefficients are multilinear polynomials over $C$ of degree $T$ in $x$.

State of the system after query $t$, on input $x$ is written $|\psi^x_t\rangle$

Initial state: $|0\rangle \otimes |0\rangle \otimes |0\rangle \in Q \otimes B \otimes Z$

After applying $U_0$:

$$\sum_{i,b,z} \alpha^0_{i,b,z} |i\rangle \otimes |b\rangle \otimes |z\rangle \in Q \otimes B \otimes Z$$

After query $O_x$:

$$\sum_{i,b,z} (1-x_i)\alpha^0_{i,b,z} + x_i \alpha^0_{i,b,z} |i\rangle \otimes |b\rangle \otimes |z\rangle$$

$$\sum_{i,b,z} \alpha^1_{i,b,z}(x)$$

Constant (degree 0)

degree 1
Express the final state on input x as a superposition whose coefficients are multilinear polynomials of degree T in x.

Applying $U_t$ does not change the degree.

After each query, degree increases by 1:

$$
\sum_{i,b,z} \left((1-x_i)\alpha^t_{i,b,z}(x) + x_i\alpha^t_{i,b,z}(x)\right)|i\rangle\otimes|b\rangle\otimes|z\rangle
$$

$x_i^2 = x_i$, $x_i \in \{0, 1\}$

At the end, degree of each variable is 1 and total degree is T
Express the final state on input $x$ as a superposition whose coefficients are multilinear polynomials of degree $2T$ in $x$.

At the end,

$$|\psi^x_T\rangle = \sum_{i,b,z} \alpha_{i,b,z}^T(x) |i\rangle \otimes |b\rangle \otimes |z\rangle$$

- $\alpha_{i,b,z}^T(x)$: polynomial over $\mathbb{C}$, multilinear of degree $T$
- Acceptance probability $P(x) = \sum_{i,b,z: \text{accept}} |\alpha_{i,b,z}^T(x)|^2$ is a multilinear polynomial over $\mathbb{R}$ of degree $2T$
Lemma  Acceptance probability $P(x) = \sum_{i,b,z: \text{accept}} |\alpha_{i,b,z}^T(x)|^2$ is a multilinear polynomial over $\mathbb{R}$ of degree $2T$

Definition  For any boolean function $f$, a polynomial $q$ represents $f$ if
$$\forall x \in \{0, 1\}^n f(x) = q(x)$$

Proposition  For any boolean function $f$, the representing polynomial $q$ is unique

Definition  $\deg(f')$ is the degree of polynomial $q$ representing $f$.

Theorem  For any boolean function $f$, $Q_E(f) \geq \deg(f')/2$
Theorem  For any boolean function $f$, $Q_E(f) \geq \deg(f)/2$ 

Example 1  $\text{PARITY}(x_1 \cdots x_n) = \sum_{i=1}^{n} x_i \pmod{2}$ 

$$q_\oplus(x_1 \cdots x_n) = \frac{1}{2} \left( 1 - \prod_{i=1}^{n} (1 - 2x_i) \right)$$ 

$Q_E(\text{PARITY}) \geq n/2$
Theorem For any boolean function $f$, $Q_E(f) \geq \deg(f)/2$

Example 2 \[ \text{OR}(x_1 \cdots x_n) = \exists i \ x_i = 1 \]
\[ q \sqrt{x_1 \cdots x_n} = 1 - \prod_{i=1}^{n} (1 - x_i) \]

$Q_E(\text{PARITY}) \geq n/2$
2-sided error quantum query complexity

Recall: Acceptance probability $P(x) = \sum_{i,b,z: \text{accept}} |\alpha_{i,b,z}^T(x)|^2$

is a multilinear polynomial over $\mathbb{R}$ of degree $2T$

Definition For any boolean function $f'$, a polynomial $q \in \epsilon$-approximates $f'$ if

$$\forall x \in \{0, 1\}^n, |f(x) - q(x)| \leq \epsilon$$

NB The approximating polynomial $q$ is not necessarily unique

Definition $\tilde{\deg}(f)$ is the minimum degree of any polynomial $q$ $1/3$-approximating $f'$.

Theorem For any boolean function $f$, $Q_2(f) \geq \tilde{\deg}(f)/2$
III. Adversary method

- Evaluate the difference between the final state on inputs that evaluate to 0 and inputs that evaluate to 1.
- Deterministic lower bounds (decision trees).
- Quantum lower bounds
- Examples
### Summary of adversary lower bounds

<table>
<thead>
<tr>
<th>Method</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deterministic method</strong></td>
<td>$DT(f) \geq \frac{</td>
</tr>
<tr>
<td>- General statement</td>
<td></td>
</tr>
<tr>
<td>- Combinatorial bound</td>
<td>$\geq \max\left{ \frac{m}{l}, \frac{m'}{l'} \right}$</td>
</tr>
<tr>
<td><strong>Quantum method</strong></td>
<td>$Q_\varepsilon(f) \geq \min_{p^x,p^y} \max_{x,y: f(x) \neq f(y)} \frac{c_\varepsilon}{2} \frac{1}{\sum_i \sqrt{p^x(i)p^y(i)}}$</td>
</tr>
<tr>
<td>- Minimax method</td>
<td></td>
</tr>
<tr>
<td>- General size method</td>
<td>$Q_\varepsilon(f) \geq \frac{c_\varepsilon</td>
</tr>
<tr>
<td>- Combinatorial bound [A02]</td>
<td>$\geq \frac{c_\varepsilon \sqrt{mm'}}{2 \sqrt{ll'}}$</td>
</tr>
<tr>
<td><strong>Weighted method [A03, LM04]</strong></td>
<td>$Q_\varepsilon(f) \geq \frac{c_\varepsilon}{2} \min_{x,y,i} \sqrt{p(x)p(y)p'<em>{x,i}(y)p'</em>{y,i}(x)} \frac{1}{q(x,y)}$</td>
</tr>
<tr>
<td><strong>Spectral method [BSS03]</strong></td>
<td>$Q_\varepsilon(f) \geq \frac{c_\varepsilon}{2 \max_i |\Gamma_i|}$</td>
</tr>
</tbody>
</table>
Combinatorial part: size of the largest piece
Adversary method, deterministic setting

Goal: separate \( \{x : f(x) = 1\} \) from \( \{y : f(y) = 0\} \) with minimum queries to input bits

- Subrelation \( R_i \): pairs for which query \( i \) is useful
- Queries \( \geq \) number of \( R_i \) needed to cover \( R \).

\[
DT(f) \geq \frac{|R|}{\max_i |R_i|}
\]
Deterministic lower bound

- **left degree** of $R \geq m$
  
  **right degree** $\geq m'$

- **left degree** of all $R_i \leq l$

  **right degree** $\leq l'$

- $|R| \geq m |X|, m' |Y|$

- $|R_i| \leq l |X|, l' |Y|$

\[
DT(f) \geq \frac{|R|}{\max_i |R_i|} \geq \max\left\{ \frac{m}{l}, \frac{m'}{l'} \right\}
\]
Progress on $x, y, i$ after a query

At time $t$, progress towards distinguishing $(x, y) \in R_i$ by making one query:

- **Deterministic case**
  $$DProgress_{t}^{x,y}(i) = \begin{cases} 1 & \text{if query}(x, t) = i \\ 0 & \text{otherwise} \end{cases}$$

- **Randomized case**
  $$RProgress_{t}^{x,y}(i) = 2 \min \{ p_{t}^{x}(i), p_{t}^{y}(i) \}$$

- **Quantum case**? (next slide)
Quantum progress

State of the system after query $t$, on input $x$ is written $|\psi_t^x\rangle$

Fix $(x,y) \in R$.

- At beginning, $\langle \psi_0^x | \psi_0^y \rangle = 1$
- At each time step,
  $$|\langle \psi_t^x | \psi_t^y \rangle - \langle \psi_{t+1}^x | \psi_{t+1}^y \rangle| \leq 2 \sum_{i: x_i \neq y_i} \sqrt{p_t^x(i)p_t^y(i)}$$
- At the end, $|\langle \psi_T^x | \psi_T^y \rangle| \leq 2\sqrt{\varepsilon(1-\varepsilon)}$

$$T \cdot \sum_i 2\sqrt{p_t^x(i)p_t^y(i)} \geq 1 - 2\sqrt{\varepsilon(1-\varepsilon)}$$
Quantum progress

State of the system after query $t$, on input $x$ is written $|\psi^x_t\rangle$

Fix $(x, y) \in R$.

- At beginning, $\langle \psi^x_0 | \psi^y_0 \rangle = 1$
- At each time step,
  $$|\langle \psi^x_t | \psi^y_t \rangle - \langle \psi^x_{t+1} | \psi^y_{t+1} \rangle| \leq 2 \sum_{i : x_i \neq y_i} \sqrt{p^x_t(i)p^y_t(i)}$$
- At the end,
  $$|\langle \psi^x_T | \psi^y_T \rangle| \leq 2 \sqrt{\epsilon(1 - \epsilon)}$$

$$T \cdot \sum_i 2\sqrt{p^x_t(i)p^y_t(i)} \geq 1 - 2 \sqrt{\epsilon(1 - \epsilon)}$$

$Progress^t_{x,y}(i)$

Amplitude$^2$ of $|i\rangle$ in query $t$, input $x$
State of the system after query $t$, on input $x$ is written $|\psi^x_t\rangle$

Fix $(x, y) \in R$.

- At beginning, $\langle \psi_0^x | \psi_0^y \rangle = 1$
- At each time step,
  $|\langle \psi^x_t | \psi^y_t \rangle - \langle \psi^x_{t+1} | \psi^y_{t+1} \rangle| \leq 2 \sum_{i: x_i \neq y_i} \sqrt{p^x_t(i)p^y_t(i)}$
- At the end, $|\langle \psi^x_T | \psi^y_T \rangle| \leq 2\sqrt{\varepsilon(1 - \varepsilon)}$

$T \cdot \sum_{i} \left( 2 \sqrt{p^x_t(i)p^y_t(i)} \right) \geq 1 - 2\sqrt{\varepsilon(1-\varepsilon)}$ 

$\text{Progress}^t_{x,y}(i)$
Quantum progress on $x, y, i$ after a query

At time $t$, progress towards distinguishing $(x, y) \in R_i$ by making one query:

- **Deterministic case**
  
  \[
  DProgress^{x,y}_t(i) = \begin{cases} 
  1 & \text{if } \text{query}(x, t) = i \\
  0 & \text{otherwise}
  \end{cases}
  \]

- **Randomized case**
  
  \[
  RProgress^{x,y}_t(i) = 2 \min\{p^x_t(i), p^y_t(i)\}
  \]

- **Quantum case**
  
  \[
  QProgress^{x,y}_t(i) = 2\sqrt{p^x_t(i)p^y_t(i)}
  \]
“Minimax method”

- We have shown that for any given quantum algorithm for $f$,
  $$\forall x, y, f(x) \neq f(y) \quad Q_\varepsilon(f) \geq c_\varepsilon \frac{1}{2} \sum_{i:x_i \neq y_i} \sqrt{p_t^x(i)p_t^y(i)}$$

- Minimax method:
  $$Q_\varepsilon(f) \geq \min_{p^x, p^y} \max_{f(x) \neq f(y)} c_\varepsilon \frac{1}{2} \sum_{i:x_i \neq y_i} \sqrt{p^x(i)p^y(i)}$$

where $\{p^x\}, \{p^y\}$ are families of probability distributions over $[n]$
General bound for quantum query complexity

Goal: lower bound in terms of overall progress on $R_i$ after a query:

$$Progress_t(i) := \sum_{x,y} Progress^x_y(i)$$

$$c_\varepsilon \leq T \cdot \sum_i Progress^x_y(i)$$

$$\sum_{(x,y) \in R} c_\varepsilon \leq T \cdot \sum_i Progress_t(i)$$

$$Q_\varepsilon(f) \geq \frac{c_\varepsilon |R|}{\sum_i Progress_t(i)}$$
Ambainis’ unweighted method

Claim: \[ \sum_{i} Progress_t(i) \leq 2 \sqrt{l |X| \cdot l' |Y|} \]
Ambainis’ unweighted method

Claim: \[ \sum_{i} Progress_t(i) \leq 2 \sqrt{l|X| \cdot l'|Y|} \]

Proof: \[ \sum_{i} Progress_t(i) \]

\[ = \sum_{x,y,i} Progress_t^{x,y}(i) \]

\[ = 2 \sum_{x,y,i} \sqrt{p_t^x(i)p_t^y(i)} \]

\[ \leq 2 \sqrt{\sum_{x} \sum_{i} \sum_{y} p_t^x(i) \sum_{y} \sum_{i} \sum_{x} p_t^y(i)} \]

\[ \leq 2 \sqrt{l|X| \cdot l'|Y|} \]
Ambainis’ unweighted method

Claim: \( \sum_i \text{Progress}_t(i) \leq 2\sqrt{l|X| \cdot l'|Y|} \)

Proof: \( \sum_i \text{Progress}_t(i) \)

\[
= \sum_{x,y,i} \text{Progress}^{x,y}_t(i) \\
= 2 \sum_{x,y,i} \sqrt{p^x_t(i)p^y_t(i)} \\
\leq 2 \sqrt{\sum_x \sum_i \sum_y p^x_t(i) \sum_y \sum_i \sum_x p^y_t(i)} \leq 2 \sqrt{l|X| \cdot l'|Y|}
\]
Ambainis’ unweighted method

Claim: \( \sum_{i} Progress_{t}(i) \leq 2 \sqrt{l |X| \cdot l' |Y|} \)

Proof: \( \sum_{i} Progress_{t}(i) \)

\[
= \sum_{x, y, i} Progress_{t}^{x, y}(i) \\
= 2 \sum_{x, y, i} \sqrt{p_{x}^{t}(i) p_{y}^{t}(i)} \\
\leq 2 \sqrt{\sum_{x} \sum_{i} \sum_{y} p_{x}^{t}(i) \sqrt{\sum_{y} \sum_{i} \sum_{x} p_{y}^{t}(i)}} \\
\leq 2 \sqrt{l |X| \cdot l' |Y|}
\]
Ambainis’ unweighted method

Claim: \[ \sum_i Progress_t(i) \leq 2 \sqrt{l|X| \cdot l'|Y|} \]

Proof: \[ \sum_i Progress_t(i) \]

\[ = \sum_{x,y,i} Progress_t^{x,y}(i) \]

\[ = 2 \sum_{x,y,i} \sqrt{p_t^x(i)p_t^y(i)} \]

\[ \leq 2 \sqrt{\sum_x \sum_i \sum_y p_t^x(i) \sum_y \sum_i \sum_x p_t^y(i)} \leq l \text{ terms} \]

\[ \leq 2 \sqrt{l|X| \cdot l'|Y|} \]
Ambainis’ unweighted method

Claim: \( \sum_{i} Progress_t(i) \leq 2\sqrt{l|X| \cdot l'|Y|} \)

Proof: \( \sum_{i} Progress_t(i) \)

\[
\begin{align*}
&= \sum_{x,y,i} Progress^x,y_t(i) \\
&= 2 \sum_{x,y,i} \sqrt{p^x_t(i)p^y_t(i)} \\
&\leq 2 \sqrt{\sum_{x} \sum_{i} \sum_{y} p^x_t(i)} \sqrt{\sum_{y} \sum_{i} \sum_{x} p^y_t(i)} \\
&\leq 2 \sqrt{l|X| \cdot l'|Y|}
\end{align*}
\]
Ambainis’ unweighted method

Claim: \[ \sum_i \text{Progress}_t(i) \leq 2\sqrt{l|X| \cdot l'|Y|} \]

Corollary: \[ Q_{\varepsilon}(f) \geq \frac{c_\varepsilon |R|}{\sum_i \text{Progress}_t(i)} \geq \frac{c_\varepsilon \sqrt{|X| \cdot |X'| \cdot |Y| \cdot |Y'|}}{2\sqrt{|X| \cdot |X'| \cdot |Y| \cdot |Y'|}} \]

\[ Q_{\varepsilon}(f) \geq \frac{c_\varepsilon}{2} \sqrt{\frac{mm'}{ll'}} \]
Example: OR function

- Deterministic queries
  \[ \geq \max\{m/l, m'/l'\} \]
  \[ = n \]

- Quantum queries
  \[ \geq \sqrt{\frac{mm'}{ll'}} \]
  \[ = \sqrt{n} \]
Unweighted \(\ell_{\text{max}}\) lower bound

- **left degree of** \(R \geq m\)
  - right degree \(\geq m'\)
- **left degree of** \(R_i \leq l_i\)
  - right degree \(\leq l'_i\)
- \(|R| \geq m \ |X|, m' \ |Y|\)

\[
\sum_i \text{Progress}_t(i) \leq 2 \sqrt{\max_i \{l_i \ |X| \cdot l'_i \ |Y|\}}
\]

\[
Q_\varepsilon(f) \geq \frac{c_\varepsilon}{2} \sqrt{\frac{mm'}{\max_i \{l_i l'_i\}}}
\]
Example: Connectivity

- **Deterministic queries**
  \[ \geq \max \{ \frac{m}{l}, \frac{m'}{l'} \} \]
  \[ = n^2 \]

- **Quantum queries**
  \[ \geq \sqrt{\frac{mm'}{ll'}} \]
  \[ = n^{3/2} \]

- Example: Connectivity

\[ m \approx n^2 \quad m' \approx n^2 \]
\[ (x) \quad l_i \approx n \quad l_i' = 1 \]
\[ (l) \quad l_i \approx 1 \quad l_i' = n \]
IV. Weighted adversary method

• Weights added to rebalance the combinatorial structure of the function

• Quantum lower bound

• Example
Weighted method
Weight scheme for adversary method

\[
\sum_{x,y \in R} q(x,y) = 1, \quad \sum_{x \in X} p(x) = 1, \quad \sum_{y \in Y} p'_{x,i}(y) = 1
\]
Consider three distributions over pairs \((x,y) \in R\)

\[
\begin{align*}
P(x,y) &= \sum_i p(x)p_t^x(i)p_{x,i}(y), \\
P'(x,y) &= \sum_i p(y)p_t^y(i)p'_{y,i}(x), \\
Q(x,y) &= q(x,y)
\end{align*}
\]

Claim: \(\exists (x,y) \in R \sum_i \text{Progress}_t^{x,y}(i)\)

\[
\leq 2\max_i \frac{q(x,y)}{\sqrt{p(x)p'_{x,i}(y)p(y)p'_{y,i}(x)}}
\]
Consider three distributions over pairs \((x, y) \in R\):

\[
P(x, y) = \sum_i p(x)p^x_t(i)p^y_{x,i}(y),
\]

\[
P'(x, y) = \sum_i p(y)p^y_t(i)p^y_{y,i}(x),
\]

\[
Q(x, y) = q(x, y)
\]

Claim: \(\exists (x, y) \in R \sum_i \text{Progress}^{x,y}_t(i)\)

\[
\leq 2\max_i \frac{q(x, y)}{\sqrt{p(x)p'_{x,i}(y)p(y)p'_{y,i}(x)}}
\]

Proof: \(\exists x, y \sqrt{P(x, y)P'(x, y)} \leq Q(x, y)\)

\[
\exists x, y \sum_i \sqrt{p^x_t(i)p^y_t(i)} \leq \frac{q(x, y)}{\min_i \sqrt{p(x)p(y)p'_{x,i}(y)p'_{y,i}(x)}}
\]

\[
\sum_{x, y} \sqrt{P(x, y)P'(x, y)} \leq \sqrt{\sum_{x, y} P(x, y) \sum_{x, y} P'(x, y)} \leq 1 = \sum_{x, y} Q(x, y)
\]
Consider three distributions over pairs \((x, y) \in R\)

\[
\begin{align*}
P(x, y) &= \sum_i p(x)p^x_t(i)p'_{x, i}(y), \\
P'(x, y) &= \sum_i p(y)p^y_t(i)p'_{y, i}(x), \\
Q(x, y) &= q(x, y)
\end{align*}
\]

Claim: \(\exists (x, y) \in R \sum_i \text{Progress}^x_{t, y}(i)\)

\[
\leq 2\max_i \frac{q(x, y)}{\sqrt{p(x)p^x_{x,i}(y)p(y)p'_{y,i}(x)}}
\]

Corollary:
Recall that \(\forall (x, y) \in R\) \(Q_\varepsilon(f) \geq \frac{c_\varepsilon}{\sum_i \text{Progress}^x_{t, y}(i)}\)

\[
Q_\varepsilon(f) \geq \frac{c_\varepsilon}{2} \min_{x, y, i} \frac{\sqrt{p(x)p(y)p^x_{x,i}(y)p'_{y,i}(x)}}{q(x, y)}
\]
Example: Ambainis’ function

\[
f(x_1x_2x_3x_4) = \begin{cases} 
1 & \text{if } x_1 \leq x_2 \leq x_3 \leq x_4 \\
1 & \text{if } x_1 \geq x_2 \geq x_3 \geq x_4 \\
0 & \text{otherwise}
\end{cases}
\]

\[
Q_{\varepsilon}(f) \geq \frac{c_\varepsilon}{2} \min_{x,y,i} \sqrt{\frac{p(x)p(y)p'_{x,i}(y)p'_{y,i}(x)}{q(x,y)}}
\]
Example: Ambainis’ function

\[ f(x_1 x_2 x_3 x_4) = \begin{cases} 
1 & \text{if } x_1 \leq x_2 \leq x_3 \leq x_4 \\
1 & \text{if } x_1 \geq x_2 \geq x_3 \geq x_4 \\
0 & \text{otherwise}
\end{cases} \]
Example: Ambainis’ function

\[ f(x_1 x_2 x_3 x_4) = \begin{cases} 
1 & \text{if } x_1 \leq x_2 \leq x_3 \leq x_4 \\
1 & \text{if } x_1 \geq x_2 \geq x_3 \geq x_4 \\
0 & \text{otherwise}
\end{cases} \]

\[ Q_\varepsilon(f) \geq \frac{c_\varepsilon}{2 \min_{x,y,i}} \frac{\sqrt{p(x)p(y)p'_{x,i}(y)p'_{y,i}(x)}}{q(x,y)} \]
Example: Ambainis’ function

\[
f(x_1x_2x_3x_4) = \begin{cases} 
1 & \text{if } x_1 \leq x_2 \leq x_3 \leq x_4 \\
1 & \text{if } x_1 \geq x_2 \geq x_3 \geq x_4 \\
0 & \text{otherwise}
\end{cases}
\]

\[Q_{\varepsilon}(f) \geq \frac{c_{\varepsilon}}{2} \min_{x,y,i} \frac{\sqrt{p(x)p(y)p'_{x,i}(y)p'_{y,i}(x)}}{q(x,y)}\]
Example: Ambainis’ function

\[ f(x_1 x_2 x_3 x_4) = \begin{cases} 
1 & \text{if } x_1 \leq x_2 \leq x_3 \leq x_4 \\
1 & \text{if } x_1 \geq x_2 \geq x_3 \geq x_4 \\
0 & \text{otherwise}
\end{cases} \]

\[ Q_\varepsilon(f) \geq \frac{c_\varepsilon}{2 \min_{x,y,i}} \sqrt{p(x)p(y)p'_{x,i}(y)p'_{y,i}(x)} / q(x,y) = \frac{20}{8} = \frac{5}{2} \]
V. Certificates and limits

- Certificate complexity
- Upper bound on adversary method in terms of certificate complexity
Certificate complexity

Let $f$ be a boolean function over $n$ variables, and let $w$ in $\{0, 1, *\}^n$ be a partial assignment to $x_1 \ldots x_n$, where $w_i = *$ means variable is unassigned. $\text{Size}(w)$ is the number of assigned (non-*) variables in $w$

- $w$ is a 0-certificate for $f$ if $f(x) = 0$ for any assignment $x$ in $\{0, 1\}^n$ that agrees with $w$ on all assigned (non-*) variables.

- $w = ****1****$ is a 1-certificate for OR
- $w = 00000000$ is a 0-certificate for OR
- $w = *****0****$ is a 0-certificate for AND
- $w = 111111111$ is a 1-certificate for AND

- $C_0(f) = \min \{\text{size}(w) : w$ is a 0 certificate for $f\}$.
- $C_1(\text{OR}) = 1$, $C_0(\text{OR}) = n$
Limits of Minimax (adversary) method

Recall the Minimax method:

\[
Q_{\varepsilon}(f) \geq \min_{p^x, p^y} \max_{f(x) \neq f(y)} \frac{c_{\varepsilon}}{2} \sum_{i : x_i \neq y_i} \sqrt{p^x(i)p^y(i)} \geq \frac{1}{n^{1/2}} \sum_{\text{i : size(w)}} \geq \sqrt{nC_0(f)}
\]

Let \( w \) be a 0-certificate for \( f \), and \( x, y \) such that \( f(x) = 0, f(y) = 1 \)

Define

\[
p^x(i) = \begin{cases} \frac{1}{\text{size}(w)} & \text{if } w_i \neq 0 \\ 0 & \text{otherwise} \end{cases}
\]

\[
p^y(i) = \frac{1}{n}
\]

then

\[
\sum_{i : x_i \neq y_i} \sqrt{p^x(i)p^y(i)} \geq \sqrt{\frac{1}{n} \frac{1}{\text{size}(w)}} \geq \sqrt{nC_0(f)}
\]

so

\[
\text{Minimax}(f) \leq \sqrt{nC_0(f)}
\]
Limits of Minimax method

\[ Q_\varepsilon(f) \geq \min_{p^x,p^y} \max_{x,y \in \{x,y\}} \frac{c_\varepsilon}{2} \sum_{i: x_i \neq y_i} \frac{1}{\sqrt{p^x(i)p^y(i)}} \]

For all boolean functions, we showed

\[ \text{Minimax}(f) \leq \sqrt{nC_0(f)}, \sqrt{nC_1(f)} \]

For total boolean functions,

\[ \text{Minimax}(f) \leq \sqrt{C_0(f)C_1(f)} \]

For the polynomial method:

\[ \deg(f) \leq C_0(f)C_1(f) \]
VI. Polynomial vs adversary method

- Cases when adversary method is weak
- Example when the adversary method is stronger than the polynomial method
Adversary method is weak if...

\[ Q_\varepsilon(f) \geq \min_{p^x, p^y} \max_{x \neq y} \frac{c_\varepsilon}{2} \frac{1}{\sqrt{p^x(i)p^y(i)}} \]

Promise problems where Hamming distance between 0 and 1 instances is large, and/or certificates are small

**Example:** Element distinctness. Given a function \([n] \rightarrow [n]\), decide if function is injective.

Lower bound is \(n^{2/3}\) (Aaronson & Shi 04)

Certificate size: \(\text{Minimax}(f) \leq \sqrt{C_0(f)C_1(f)}\)

\(C_{\text{collision}} = 2\)

\(C_{\text{injective}} = n\)

Adversary method cannot prove a lower bound better than \(\sqrt{n}\)
Example where polynomial method is weaker

Ambainis’ function: $p(x)$ has degree 2

$$p(x) = x_1 x_2 + x_2 x_3 + x_3 x_4 - x_1 - x_4$$

Adversary method: 2.5

By composing the function, gap is amplified:

polynomial bound: $2^n$

Adversary method $2.5^n$
## Polynomial for Ambainis’ function

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_1x_2$</th>
<th>$x_2x_3$</th>
<th>$x_3x_4$</th>
<th>$x_1x_4$</th>
<th>$x_1x_2+x_2x_3$</th>
<th>$-x_3x_4$</th>
<th>$1-x_2-x_3$</th>
<th>sorted($x$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
VII. Spectral method

- Statement and idea
- Equivalence with weighted methods
Spectral method

Nonnegative weights matrix

\[ \Gamma[x, y] = \begin{cases} 0 & \text{if } f(x) = f(y) \\ \Gamma[x, y] & \text{otherwise} \end{cases} \]

\[ \|\Gamma\| = \max_{|u|=|v|=1} |u^* \Gamma v| \]

\[ Q_\varepsilon(f) \geq \frac{c_\varepsilon}{2} \frac{\|\Gamma\|}{\max_i \|\Gamma_i\|} \]

Idea: \( q(x, y) \) derived from
\( p(x), p(y) \) derived from \( u, v \) maximizing \( |u^* T v| \)
\( p'_{y,i}(x), p'_{x,i}(y) \) derived from \( u_i, v_i \) maximizing \( |u_i^* T_i v_i| \)
Equivalence of adversary methods

- Semidefinite programming [BSS03]
- Minimax method [LM04]
- Spectral method [BS02, BSS03]
- Weighted adversary [Am03, Aa04]
- Unweighted adversary [Am02]
- Polynomial method [BBCMW]

[ŠS04] All these methods are equivalent
VIII. Formula size lower bounds

- Boolean circuits and formulas
- Khrapchenko’s method
Boolean formula size

Best lower bound: $4.5 \cdot n$
[Lachish Raz 01, Iwama Morizumi 02]

$L(f)$ = number of leaves in the formula
Best lower bound: $n^{3-o(1)}$
[Håstad 98]
Khrapchenko’s method (1971)

Theorem [LLS05] $L(f) \geq \text{Minimax}(f)^2 = \text{weighted adversary}(f)^2$

Follows by setting

$$q(x, y) = 1 \frac{1}{|R|}$$

$$p(x) = \frac{1}{|A|}, \quad p(y) = \frac{1}{|B|}$$

$$p'_{x,i}(y) = 1, \quad p'_{y,i}(x) = 1$$
Lower bounds for formula size

Query complexity

Minimax. spectral, weighted adversary [Am03, Aa04, BS02, BSS03, LM04, SS04]

Unweighted adversary [Am02]

Formula size

Rectangle cover (communication complexity)

(Weighted adversary)^2 [LLS05]

Weak spectral method [Koutsoupias, 93]

Random restrictions [Håstad 98]

Khrapchenko [K71]
Further reading

- **Polynomial method**

- **Adversary method**