Polynomials, quantum query complexity, and Grothendieck's inequality

Scott Aaronson[1], Andris Ambainis[2], Jānis Iraids[2], Martins Kokainis[2], Juris Smotrovs[2]

[1]Department of Computer Science, UT Austin

[2]Faculty of Computing, University of Latvia

Joint Estonian-Latvian Theory Days 2016

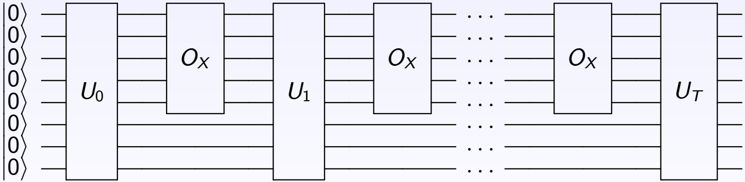
Aaronson, Ambainis, Iraids, K, Smotrovs,
Polynomials, Quantum Query Complexity, and Grothendieck's Inequality,
arXiv:1511.08682

Query model

- Function $f(x_1, x_2, \ldots, x_n)$, $x_i \in \{0, 1\}$.
- x_i given by a black box:

- Complexity $=$ number of queries.

Quantum query model



- $U_0, U_1, \ldots, U_T$, independent of $x_1, \ldots, x_n$.
- O_X – query operators:

$$\sum_i a_i \left|i\right\rangle \xrightarrow{O_X} \sum_i a_i (-1)^{x_i} \left|i\right\rangle$$

- $Q_\epsilon(f)$ – minimum number of queries in a quantum algorithm computing f correctly with probability $\geq 1-\epsilon$.

Quantum algorithms that make T queries	$\Longrightarrow$ [BBCMW01]	Multilinear polynomials of degree $2T$

- Lower bounds on quantum query complexity
  - OR: no polynomial of degree $o(\sqrt{n})$ approximating OR [NS94], thus no quantum algorithm making $o(\sqrt{n})$ queries.
  - Collision problem, element distinctness problem, ...
- The obtained bounds can be asymptotically lower than $Q_\epsilon(f)$.
- Opposite direction?

Multilinear polynomials of degree d

$\Longrightarrow$ [BBCMW01]

Quantum algorithms that make $O(d^6)$ queries

A multilinear polynomial of degree d

& [ABK16]

Quantum algorithms make $\tilde{\Omega}(d^4)$ queries

Quantum algorithms that make T queries

$\Longrightarrow$

$\overset{??}{\Longleftarrow}$

Multilinear polynomials of degree $2T$

Quantum algorithms that make T queries $\implies$ Multilinear polynomials of degree $2T$

Quantum algorithms that make T queries $\overset{??}{\impliedby}$ Multilinear polynomials of degree $2T$

This work:

Quantum algorithms that make 1 query $\iff$ Multilinear polynomials of degree 2

- Recently shown [AA15]:
  - A task that requires 1 query quantumly and $\Theta(\sqrt{n})$ queries classically.
  - Any quantum algorithm which makes 1 query can be simulated by a probabilistic algorithm making $O(\sqrt{n})$ queries.

Multilinear polynomials

- $p : \mathbb{R}^n \to \mathbb{R}$ is a multilinear polynomial of degree d if

$$p(x_1, \ldots, x_n) = \sum_{\substack{S \subset [n] \\ |S| \leq d}} a_S \prod_{i \in S} x_i, \qquad x_j \in \mathbb{R}.$$

A multilinear polynomial $p : \mathbb{R}^n \to \mathbb{R}$ represents $f : \{-1, 1\}^n \to \{0, 1\}$ with error $\delta \in [0; 0.5)$ if, for all $x \in \{-1, 1\}^n$,

- $f(x) = 0 \;\Rightarrow\; p(x) \in [0; \delta]$,
- $f(x) = 1 \;\Rightarrow\; p(x) \in [1 - \delta; 1]$, and
- $p(x) \in [0; 1]$.

If $\delta = 0$, the polynomial p is said to represent f exactly.

Block-multilinear polynomials

- $q : \mathbb{R}^{d(n+1)} \to \mathbb{R}$ is a block-multilinear multilinear polynomial of degree d if

$$q(x^{(1)}, \ldots, x^{(d)}) = \sum_{i_1, i_2, \ldots, i_d = 0 \ldots n} a_{i_1 i_2 \ldots i_d} x^{(1)}_{i_1} x^{(2)}_{i_2} \ldots x^{(d)}_{i_d}, \quad x^{(j)} \in \mathbb{R}^{n+1}.$$

A block-multilinear polynomial $q : \mathbb{R}^{d(n+1)} \to \mathbb{R}$ of degree d represents $f : \{-1,1\}^n \to \{0,1\}$ with error $\delta \in [0; 0.5)$ if, for all $x \in \{-1,1\}^n$,

- $f(x) = 0 \;\Rightarrow\; q(\tilde{x}, \tilde{x}, \ldots, \tilde{x}) \in [0; \delta], \quad \tilde{x} := (1, x)$,
- $f(x) = 1 \;\Rightarrow\; q(\tilde{x}, \tilde{x}, \ldots, \tilde{x}) \in [1-\delta; 1], \quad \tilde{x} := (1, x)$, and
- $q(x^{(1)}, \ldots, x^{(d)}) \in [-1; 1]$ for all $x^{(1)}, \ldots, x^{(d)} \in \{-1,1\}^{n+1}$.

If $\delta = 0$, the polynomial q is said to represent f exactly.

Example

- Consider $NAE(x_1, x_2, x_3) = \neg(x_1 = x_2 = x_3)$.
- Ordinary exact representation:

$$p(x_1, x_2, x_3) = \frac{3 - x_1x_2 - x_1x_3 - x_2x_3}{4}$$

- Block-multilinear exact representation:

$$q(x_0, \ldots, x_3, y_0, \ldots, y_3) = \frac{2x_0y_0 - x_1y_2 - x_1y_3 - x_3y_2 + x_3y_3}{4}$$

- Notice that setting $x_0 = y_0 = 1$ and $x_i = y_i$ yields

$$q(1, x_1, x_2, x_3, 1, x_1, x_2, x_3) = p(x_1, x_2, x_3).$$

From quantum algorithms to polynomials

- $\widetilde{\deg}_\epsilon(f)$: the minimum degree of a polynomial p representing f with error ϵ;
- $\widetilde{\mathrm{bmdeg}}_\epsilon(f)$: the minimum degree of a block-multilinear polynomial q representing f with error ϵ.

Theorem ([BBCMW01])

$$\mathrm{Q}_\epsilon(f) \geq 2\widetilde{\deg}_\epsilon(f)$$

Theorem ([AA15])

$$\mathrm{Q}_\epsilon(f) \geq 2\widetilde{\mathrm{bmdeg}}_\epsilon(f)$$

Theorem

$Q_\epsilon(f) = 1$ *for some* $\epsilon < 0.5 \quad \Leftrightarrow \quad \widetilde{\deg}_\delta(f) = 2$ *for some* $\delta < 0.5$

Sketch of the proof

1. From a multilinear polynomial p to a block-multilinear polynomial q.
2. By splitting variables from q to a block-multilinear polynomial q'.
3. A quantum algorithm which estimates q' by making a single query.

Estimating a polynomial with a quantum algorithm

- A block-multilinear polynomial q of degree 2:

$$q(x_1, \ldots, x_n, y_1, \ldots, y_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i y_j.$$

- Let $A = (a_{ij})$ and suppose $U = n \cdot A$ is unitary.
- One can prepare with a single query each of the states

$$|\Psi_x\rangle = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} x_i |i\rangle, \quad |\Psi_y\rangle = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} y_j |j\rangle,$$

thus with a single query it is possible to estimate

$$\langle \Psi_x | U | \Psi_y \rangle = q(x_1, \ldots, x_n, y_1, \ldots, y_n).$$

- Still works if $\|U\| \leq C$.

Preprocessing a block-multilinear polynomial

- Have: $|q| \leq 1$, i.e.,

$$\max_{x,y\in\{-1,1\}^n} \left| \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i y_j \right| \leq 1 \quad \text{or} \quad \|A\|_{\infty\to 1} \leq 1.$$

- Need: $n\,\|A\| \leq C$.
- Solution: variable splitting.
- A variable x_i can be replaced by new variables $x_{i_1}, \ldots, x_{i_k}$ as follows:

$$x_i \longrightarrow \frac{x_{i_1} + x_{i_2} + \ldots + x_{i_k}}{k}.$$

Preprocessing a block-multilinear polynomial

- Have: $|q| \leq 1$, i.e.,

$$\max_{x,y \in \{-1,1\}^n} \left| \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i y_j \right| \leq 1 \quad \text{or} \quad \|A\|_{\infty \to 1} \leq 1.$$

- Need: $n \|A\| \leq C$.
- Solution: variable splitting.
- A variable x_i can be replaced by new variables $x_{i_1}, \ldots, x_{i_k}$ as follows:

$$x_i \longrightarrow \frac{x_{i_1} + x_{i_2} + \ldots + x_{i_k}}{k}.$$

- Another block-multilinear polynomial q' is obtained with a coefficient matrix A' of size $n' \times m'$.
- Still $|q'| \leq 1$ or $\|A'\|_{\infty \to 1} \leq 1$.
- Can we achieve $\sqrt{n'm'}\,\|A'\| \leq C$?

- Another block-multilinear polynomial q' is obtained with a coefficient matrix A' of size $n' \times m'$.
- Still $|q'| \leq 1$ or $\|A'\|_{\infty \to 1} \leq 1$.
- Can we achieve $\sqrt{n'm'}\,\|A'\| \leq C$?

Claim

For each $\delta > 0$ it is possible to split variables so that the obtained matrix A' satisfies

$$\sqrt{n'm'}\,\|A'\| \leq K + \delta,$$

where $K < 1.7823$ – Groethendieck's constant.

Key idea: splitting variables is equivalent to factorizing the matrix A.

Splitting variables ≡ splitting rows/columns of A

- Splitting a variable x_i into k new variables corresponds to splitting the ith row of A into k equal rows.

Example

- Let $q = \frac{1}{2}(x_1y_1 + x_2y_1 + x_1y_2 - x_2y_2)$, then $A = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{pmatrix}$.
- Replacing x_2 with $\frac{x_2' + x_3' + x_4'}{3}$ corresponds to ...
- ... replacing A with

$$A' = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{6} & -\frac{1}{6} \\ \frac{1}{6} & -\frac{1}{6} \\ \frac{1}{6} & -\frac{1}{6} \end{pmatrix}.$$

Splitting variables $\equiv$ splitting rows/columns of A

- Splitting a variable x_i into k new variables corresponds to splitting the ith row of A into k equal rows.

Example

- Let $q = \frac{1}{2}(x_1y_1 + x_2y_1 + x_1y_2 - x_2y_2)$, then $A = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{pmatrix}$.
- Replacing x_2 with $\frac{x_2'+x_3'+x_4'}{3}$ corresponds to ...
- ... replacing A with

$$A' = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{6} & -\frac{1}{6} \\ \frac{1}{6} & -\frac{1}{6} \\ \frac{1}{6} & -\frac{1}{6} \end{pmatrix}.$$

Splitting variables $\equiv$ splitting rows/columns of A

- Splitting a variable x_i into k new variables corresponds to splitting the ith row of A into k equal rows.

Example

- Let $q = \frac{1}{2}\left(x_1y_1 + x_2y_1 + x_1y_2 - x_2y_2\right)$, then $A = \left(\begin{smallmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{smallmatrix}\right)$.
- Replacing x_2 with $\frac{x_2'+x_3'+x_4'}{3}$ corresponds to ...
- ... replacing A with

$$A' = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{6} & -\frac{1}{6} \\ \frac{1}{6} & -\frac{1}{6} \\ \frac{1}{6} & -\frac{1}{6} \end{pmatrix}.$$

- Suppose that A is of size $n \times m$ and its
  - 1st row is split into k_1 rows,
  - 2nd row – into k_2 rows,

    ...
  - nth row – into k_n rows,

  obtaining A' of size $n' \times m'$.
- Clearly, $m' = m$, $n' = k_1 + k_2 + \ldots + k_n$.
- What about $\|A'\|$?

- We have $\|A'\| = \|B\|$, where

$$B = \begin{pmatrix} \frac{a_{11}}{\sqrt{k_1}} & \frac{a_{12}}{\sqrt{k_1}} & \cdots & \frac{a_{1m}}{\sqrt{k_1}} \\ \frac{a_{21}}{\sqrt{k_2}} & \frac{a_{22}}{\sqrt{k_2}} & \cdots & \frac{a_{2m}}{\sqrt{k_2}} \\ & & \ddots & \\ \frac{a_{n1}}{\sqrt{k_n}} & \frac{a_{n2}}{\sqrt{k_n}} & \cdots & \frac{a_{nm}}{\sqrt{k_n}} \end{pmatrix}$$

- Consequently,

$$\left\|A'\right\| \sqrt{n' m'} = \|B\| \, \|w\| \, \|v\| \, ,$$

where $w = (\sqrt{k_1}, \ldots, \sqrt{k_n})$, $v = (1, \ldots, 1)$.

Splitting rows/columns $\equiv$ factorizing A

- Let A be of size $n \times m$ and $C > 0$.
- Claim:

$\exists B \in \mathbb{R}^{n \times m}$ and $w \in \mathbb{R}^n_+$, $v \in \mathbb{R}^m_+$:

- $a_{ij} = w_i b_{ij} v_j, \quad \forall i, j,$
- $w_i^2, v_j^2 \in \mathbb{Q}$, $\forall i, j,$
- $\|B\| \, \|w\| \, \|v\| = C$

$\Longleftrightarrow$

$\exists A' \in \mathbb{R}^{n' \times m'}$:

- $A \longrightarrow A'$,
- $\|A'\| \sqrt{n' m'} = C$

Splitting rows/columns $\equiv$ factorizing A

- Let A be of size $n \times m$ and $C > 0$.
- Claim:

$\exists B \in \mathbb{R}^{n \times m}$ and $w \in \mathbb{R}_+^n$, $v \in \mathbb{R}_+^m$:

- $a_{ij} = w_i b_{ij} v_j, \quad \forall i, j,$
- ~~$w_i^2, v_j^2 \in \mathbb{Q}, \forall i, j,$~~
- $\|B\| \, \|w\| \, \|v\| = C$

$\Longrightarrow$

$\forall \delta > 0$ $\exists A' \in \mathbb{R}^{n' \times m'}$:

- $A \longrightarrow A'$,
- $\|A'\| \sqrt{n' m'} = C + \delta$

Grothendieck's Inequality: I

- Suppose that
  - A is a $n \times m$ matrix with real components;
  - $\mathcal{H}$ is an arbitrary Hilbert space;
  - $\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{y}_1, \ldots, \mathbf{y}_m \in \mathcal{H}$ are of norm at most 1.

  Then

$$\left| \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} \langle \mathbf{x}_i, \mathbf{y}_j \rangle \right| \leq K \|A\|_{\infty \to 1},$$

  where

$$\|A\|_{\infty \to 1} = \max_{\substack{x \in \{-1,1\}^n \\ y \in \{-1,1\}^m}} \left| \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} x_i y_j \right|.$$

Grothendieck's Inequality: II

- Suppose that A is a $n \times n$ matrix. Then the following are equivalent:
  1. for each $\mathcal{H}$ and all $\mathbf{x}_i$, $\mathbf{y}_j \in \mathcal{H}$ (of norm ≤ 1), $i, j \in [n]$,
  $$\left| \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} \langle \mathbf{x}_i, \mathbf{y}_j \rangle \right| \leq 1;$$
  2. there is an $n \times n$ matrix B and vectors $w, v \in \mathbb{R}_+^n$, s.t.
     - $\|w\| = \|v\| = 1$;
     - $\|B\| \leq 1$;
     - $w_i b_{ij} v_j = a_{ij}$ for all i, j.

Putting everything together

- Since $\|A\|_{\infty\to 1} \leq 1$, there is a matrix B and vectors w, v s.t.

$$\|w\| = \|v\| = 1,\ \|B\| \leq K \quad \text{and} \quad w_i b_{ij} v_j = a_{ij} \text{ for all } i, j.$$

- Then we can split variables so that the obtained matrix A' satisfies $\|A'\| \sqrt{n' m'} \leq K + \delta$, for every $\delta > 0$.
- Therefore there is a 1-query quantum algorithm which estimates q' (the polynomial corresponding to A'),
- thus evaluating the polynomial q.

$\widetilde{\deg} = 2 \Leftrightarrow \widetilde{\text{bmdeg}} = 2$

Claim

Suppose that

- $p : \mathbb{R}^n \to \mathbb{R}$ *is a multilinear polynomial of degree 2,*
- $p(x) \in [0, 1]$ *for each* $x \in \{-1, 1\}^n$.

Then there exists a block-multilinear polynomial $g : \mathbb{R}^{2n+2} \to \mathbb{R}$ *s.t.*

- $\deg g = 2$,
- $g(\tilde{x}, \tilde{x}) = p(x)$, $\tilde{x} := (1, x)$, *for each* $x \in \{-1, 1\}^n$,
- $|g(z)| \leq 1$ *for each* $z \in \{-1, 1\}^{2n+2}$.

From polynomials to block-multilinear polynomials

- We have shown that $\widetilde{\deg} = 2 \Leftrightarrow \widetilde{\text{bmdeg}} = 2$. What about higher degrees?
- Generally, from each multilinear polynomial a block-multilinear one can be constructed, albeit with a larger approximation error.

Claim

Suppose that

- $p : \mathbb{R}^n \to \mathbb{R}$ *is a multilinear polynomial of degree* d,
- $|p(x)| \leq 1$ *for each* $x \in \{-1, 1\}^n$.

Then there exists a block-multilinear polynomial $g : \mathbb{R}^{d(n+1)} \to \mathbb{R}$ *s.t.*

- $\deg g = d$,
- $g(\tilde{x}, \ldots, \tilde{x}) = p(x)$ *for each* $x \in \{-1, 1\}^n$, $\tilde{x} := (1, x)$;
- $|g(z)| \leq C_d = O(3.5911...^d)$ *for each* $z \in \{-1, 1\}^{d(n+1)}$.

Key ideas:

1. replace each monomial with its symmetric block-multilinear version (average over all the ways how one could use one term per block), e.g.,

$$x_1 x_2 \dots x_r \longrightarrow \frac{1}{\binom{d}{r} r!} \sum_{\substack{B \subset [d]: \\ |B| = r}} \sum_{\substack{b: \\ b:[r] \to B \\ b \text{ -- bijection}}} x_1^{(b(1))} x_2^{(b(2))} \dots x_r^{(b(r))}.$$

2. Apply the polarization identity to show the boundedness of g:

$$d!F\left(u^{(1)}, u^{(2)}, \dots, u^{(d)}\right) = \sum_{\substack{T \subset [d] \\ T \neq \emptyset}} (-1)^{d-|T|} f\left(\sum_{j \in T} u^{(j)}\right),$$

where $f(x) := F(x, x, \dots, x)$ and $F : E^d \to \mathbb{R}$ is a d-linear and symmetric map.

- Corollary: solution of an open problem from [AA15].

Claim

Let $g : \mathbb{R}^n \to \mathbb{R}$ be a multilinear polynomial of degree d with $|g(y)| \leq 1$ for any $y \in \{-1, 1\}^n$. Then $g(y)$ can be approximated within precision $\pm\epsilon$ whp by querying $O((\frac{n}{\epsilon^2})^{1-1/d})$ variables (with a big-O constant depending on d).

- The same result (and transformation of ordinary multilinear polynomials to block-multilinear ones) has been independently shown by O'Donnell and Zhao by means of decoupling theory.

Separation between Q and bmdeg

- Q and bmdeg are not equivalent: there is a function exhibiting a quadratic separation between both measures.

Theorem

There exists f with $Q_\epsilon(f) = \tilde{\Omega}(\mathrm{bmdeg}_0^2(f))$.

- Recently [ABK16] an analogous result for Q_ϵ and $\deg_0$ using the cheat sheet framework.
- We show that the same function provides the separation between Q_ϵ and bmdeg_0.

? Characterize quantum algorithms with 2, 3, ..., queries?

? 2 queries $\equiv$ polynomials of degree 4?

Thank you for your attention!

? Characterize quantum algorithms with 2, 3, ..., queries?

? 2 queries $\equiv$ polynomials of degree 4?

Thank you for your attention!