SURVEY ON ALGEBRAIC COMPLEXITY

SRIKANTH SRINIVASAN

AARHUS UNIVERSITY

Talk Outline

1. Introduction to Algebraic Complexity
2. Lower bound Techniques
3. Polynomial Identity Testing
4. Proof Complexity
5. Possible Barriers to Lower Bounds

Talk Outline

① Introduction to Algebraic Complexity

Algebraic Complexity Theory

$(P_N(x_1, \ldots, x_N))_N$ - sequence of polynomials

$\deg(P_N) \leq N^{O(1)}$

Input: $a \in \mathbb{C}^N$ [or $\mathbb{F}^N$]

Output: $P_N(a)$

Algebraic Complexity Theory

$(P_N(x_1, \dots, x_N))_N$ - sequence of polynomials

$\deg(P_N) \le N^{O(1)}$

Input: $a \in \mathbb{C}^N$ [or $\mathbb{F}^N$]

Output: $P_N(a)$

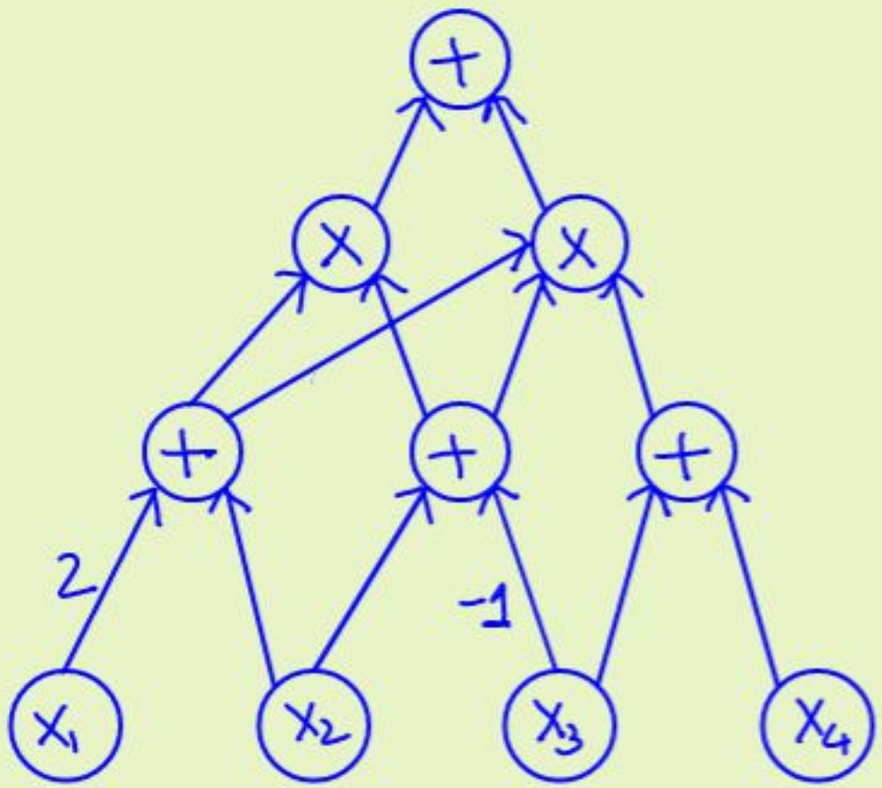
Examples: ① $\det_n\left(x_{i,j}\right)_{i,j} = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma)\, x_{1,\sigma(1)} \cdots x_{n,\sigma(n)}$

② $\operatorname{per}_n\left(x_{i,j}\right)_{i,j}$ $\quad N = n^2,\ \deg = n$

③ $IMM_{n;d} =$ 

$N \approx d \cdot n^2$, $\deg = d$

Algebraic Circuits

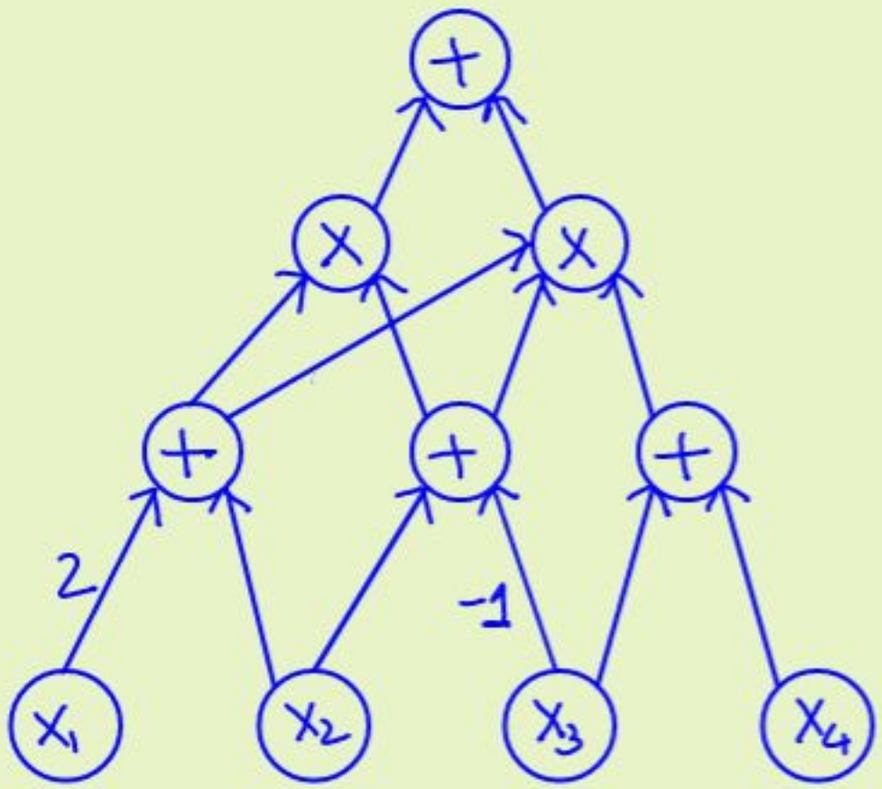


Size = # of gates = 6

Depth = length of longest path

= 3

Algebraic Circuits



Size = # of gates = 6

Depth = length of longest path
= 3

VP = sequences of polynomials with circuits of polynomial size.

Eg: $IMM_{n,d} \in VP$

$det_n \in VP$

$per_n \notin VP$?

[VP vs. VNP]

VP vs. VNP

→ Special (syntactic) case of P/poly vs. NP
[Bür]

→ Concrete approaches
  → Syntactic simplification
  → Depth-reduction

Other Syntactic Restrictions

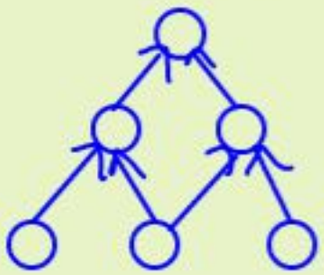
Other Syntactic Restrictions

$P(x_1, \ldots, x_N)$ – homogeneous / multilinear etc.

Other Syntactic Restrictions

$P(x_1, \ldots, x_N)$ - homogeneous / multilinear etc.

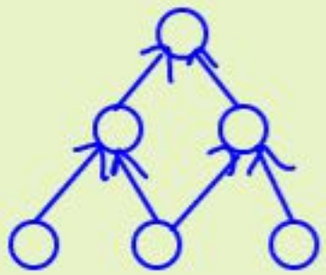
Homogeneous circuits: Each intermediate computation is homogeneous.



Other Syntactic Restrictions

$P(x_1, \ldots, x_N)$ - homogeneous / multilinear etc.

Homogeneous circuits: Each intermediate computation is homogeneous.



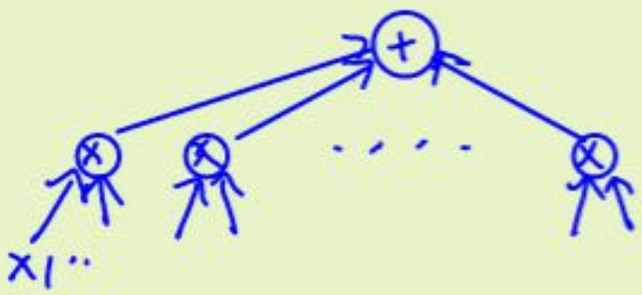
→ Sometimes, restricted circuits ≈ general circuits. [Good for lbds!]

→ Interesting situations where this may not be true.

Constant-depth Algebraic circuits

$$P(x_1, \ldots, x_N) = \sum_{e_i \geq 0 : \sum_i e_i \leq d} \alpha_{e_1, \ldots, e_N} x_1^{e_1} \cdots x_N^{e_N}$$

$\deg(P) = d$

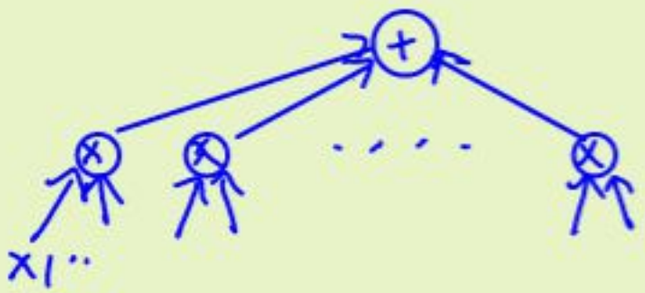


$\rightarrow$ $\Sigma\Pi$ circuit for P

Constant-depth Algebraic circuits

$$P(x_1,\dots,x_N) = \sum_{e_i \geq 0 \,:\, \sum_i e_i \leq d} \alpha_{e_1,\dots,e_N} x_1^{e_1} \cdots x_N^{e_N}$$

$\deg(P) = d$



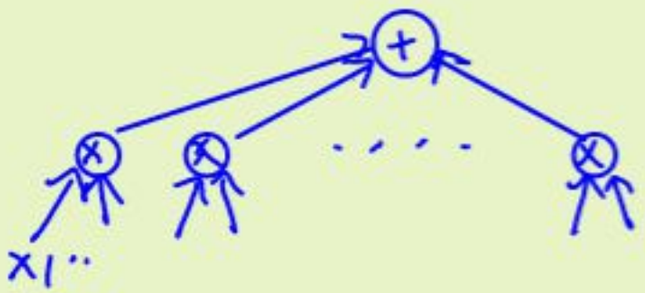
$\longrightarrow$ $\Sigma\Pi$ circuit for P

size $\approx$ # of monomials $\approx N^d$

Constant-depth Algebraic circuits

$$P(x_1, \ldots, x_N) = \sum_{e_i \geq 0 \,:\, \sum_i e_i \leq d} \alpha_{e_1, \ldots, e_N} x_1^{e_1} \cdots x_N^{e_N}$$

$\deg(P) = d$



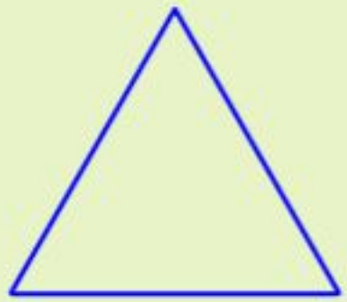
$\rightarrow$ $\Sigma\Pi$ circuit for P

size $\approx$ # of monomials $\approx N^d$

$\Sigma\Pi\Sigma$, $\Sigma\Pi\Sigma\Pi$, ...

↳ much harder to reason about!

Depth-Reduction [Brent, VSBR]

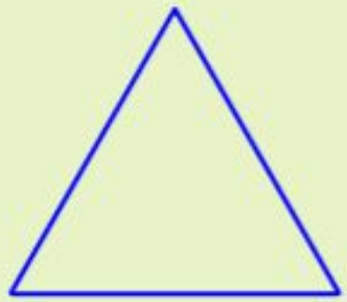


$\Longrightarrow$

General circuit $\Longrightarrow$ Const. depth circuit
poly. size subexponential size

Superpoly. lbds for general ckts $\Longleftarrow$ Exponential const. depth lower bounds

Depth-Reduction [Brent, VSBR]



$\Longrightarrow$

General circuit $\Longrightarrow$ Const. depth circuit
poly. size subexponential size

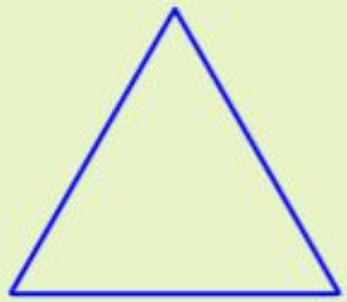
Superpoly. lbds $\Longleftarrow$ Exponential const.
for general ckts depth lower bounds

$\Sigma\Pi\Sigma\Pi$ size $N^{O(\sqrt{d})}$

Poly-sized $\Longrightarrow$ homogeneous [AV, Koi, Tav]
ckt. for

$P(x_1, \ldots, x_N)$

Depth-Reduction [Brent, VSBR]



$\Longrightarrow$

General circuit $\Longrightarrow$ Const. depth circuit
poly. size subexponential size

Superpoly. lbds $\Longleftarrow$ Exponential const.
for general ckts depth lower bounds

Poly-sized $\Nearrow$ $\Sigma\Pi\Sigma\Pi$ size $N^{O(\sqrt{d})}$
ckt. for homogeneous [AV, Koi, Tav]
$P(x_1, \ldots, x_N)$ $\Searrow$ $\Sigma\Pi\Sigma$ size $N^{O(\sqrt{d})}$
inhomogeneous [akks] (*)

Talk Outline

① Introduction to Algebraic Complexity

② Lower bound Techniques

Constant-depth circuit lower bounds

[NW]: Partial Derivative Method

→ "Complexity of $P(x_1, \ldots, x_N)$" = rank of matrix M_P

Constant-depth circuit lower bounds

[NW]: Partial Derivative Method

→ "Complexity of $P(x_1, \ldots, x_N)$" = rank of matrix M_P

→ Small circuit ⇒ small rank

→ $rk(M_1 + M_2) \leq rk(M_1) + rk(M_2)$

Example : Tensor Rank

$$\{x_1^{(1)}, \ldots, x_n^{(1)}\}, \{x_1^{(2)}, \ldots, x_n^{(2)}\}, \ldots, \{x_1^{(d)}, \ldots, x_n^{(d)}\}$$

Example : Tensor Rank

$$\{x_1^{(1)}, \dots, x_n^{(1)}\}, \{x_1^{(2)}, \dots, x_n^{(2)}\}, \dots, \{x_1^{(d)}, \dots, x_n^{(d)}\}$$

$$\underbrace{Q(\bar{x}^{(1)}, \dots, \bar{x}^{(d)})}$$

$$\sum_{i_1, \dots, i_d} \alpha_{i_1, \dots, i_d} x_{i_1}^{(1)} \cdots x_{i_d}^{(d)}$$

set-multilinear poly.

Example : Tensor Rank

$$\{x_1^{(1)}, \dots, x_n^{(1)}\}, \{x_1^{(2)}, \dots, x_n^{(2)}\}, \dots, \{x_1^{(d)}, \dots, x_n^{(d)}\}$$

$$\underbrace{Q(\bar{x}^{(1)}, \dots, \bar{x}^{(d)})}_{} = l_1^{(1)} \cdots l_1^{(d)} + \cdots + l_s^{(1)} \cdots l_s^{(d)}$$

$$\sum_{i_1, \dots, i_d} \alpha_{i_1, \dots, i_d} x_{i_1}^{(1)} \cdots x_{i_d}^{(d)}$$

set-multilinear poly.

Example : Tensor Rank

$$\{x_1^{(1)}, \dots, x_n^{(1)}\}, \{x_1^{(2)}, \dots, x_n^{(2)}\}, \dots, \{x_1^{(d)}, \dots, x_n^{(d)}\}$$

$$\underbrace{Q(\bar{x}^{(1)}, \dots, \bar{x}^{(d)})}_{\sum_{i_1, \dots, i_d} \alpha_{i_1, \dots, i_d} x_{i_1}^{(1)} \cdots x_{i_d}^{(d)}} = \underbrace{\ell_1^{(1)} \cdots \ell_1^{(d)} + \cdots + \ell_s^{(1)} \cdots \ell_s^{(d)}}_{\sum \prod \sum \text{ s.m. circuit}}$$

$\min s =$ Tensor-Rk(Q)

set-multilinear poly.

Example : Tensor Rank

$$\{x_1^{(1)}, \dots, x_n^{(1)}\}, \{x_1^{(2)}, \dots, x_n^{(2)}\}, \dots, \{x_1^{(d)}, \dots, x_n^{(d)}\}$$

$$\underbrace{Q(\bar{x}^{(1)}, \dots, \bar{x}^{(d)})}_{} = l_1^{(1)} \cdots l_1^{(d)} + \cdots + l_s^{(1)} \cdots l_s^{(d)}$$

$$\sum_{i_1, \dots, i_d} \alpha_{i_1, \dots, i_d} x_{i_1}^{(1)} \cdots x_{i_d}^{(d)}$$

set-multilinear poly.

Example : Tensor Rank

$$\{x_1^{(1)}, \dots, x_n^{(1)}\}, \{x_1^{(2)}, \dots, x_n^{(2)}\}, \dots, \{x_1^{(d)}, \dots, x_n^{(d)}\}$$

$$\underbrace{Q(\bar{x}^{(1)}, \dots, \bar{x}^{(d)})}_{} = \ell_1^{(1)} \cdots \ell_1^{(d)} + \cdots + \ell_s^{(1)} \cdots \ell_s^{(d)}$$

$$\sum_{i_1, \dots, i_d} \alpha_{i_1, \dots, i_d} x_{i_1}^{(1)} \cdots x_{i_d}^{(d)}$$

set-multilinear poly.

$$M_p = x_{i_1}^{(1)} \cdots x_{i_{d/2}}^{(d/2)} \left(\text{coeff}\left(x_{i_1}^{(1)} \cdots x_{i_d}^{(d)}\right) x_{i_{d/2+1}}^{(d/2+1)} \cdots x_{i_d}^{(d)} \right)$$

$n^{d/2} \times n^{d/2}$

Example: Tensor Rank

$$\{x_1^{(1)}, \dots, x_n^{(1)}\}, \{x_1^{(2)}, \dots, x_n^{(2)}\}, \dots, \{x_1^{(d)}, \dots, x_n^{(d)}\}$$

$$\underbrace{Q(\bar{x}^{(1)}, \dots, \bar{x}^{(d)})}_{} = \ell_1^{(1)} \cdots \ell_1^{(d)} + \cdots + \ell_s^{(1)} \cdots \ell_s^{(d)}$$

$$\sum_{i_1, \dots, i_d} \alpha_{i_1, \dots, i_d} x_{i_1}^{(1)} \cdots x_{i_d}^{(d)}$$

set-multilinear poly.

$$M_P = x_{i_1}^{(1)} \cdots x_{i_{d/2}}^{(d/2)} \Big(\text{coeff}(x_{i_1}^{(1)} \cdots x_{i_d}^{(d)}) \; x_{i_{d/2+1}}^{(d/2+1)} \cdots x_{i_d}^{(d)} \Big)$$

$n^{d/2} \times n^{d/2}$

– Columns are partial derivatives of P

Example : Tensor Rank

$$\{x_1^{(1)}, \dots, x_n^{(1)}\}, \{x_1^{(2)}, \dots, x_n^{(2)}\}, \dots, \{x_1^{(d)}, \dots, x_n^{(d)}\}$$

$$Q(\vec{x}^{(1)}, \dots, \vec{x}^{(d)}) = l_1^{(1)} \cdots l_1^{(d)} + \cdots + \underbrace{l_s^{(1)} \cdots l_s^{(d)}}_{\text{rank} \leq 1}$$

Example : Tensor Rank

$$\{x_1^{(1)}, \dots, x_n^{(1)}\}, \{x_1^{(2)}, \dots, x_n^{(2)}\}, \dots, \{x_1^{(d)}, \dots, x_n^{(d)}\}$$

$$Q(\bar{x}^{(1)}, \dots, \bar{x}^{(d)}) = l_1^{(1)} \cdots l_1^{(d)} + \cdots + \underbrace{l_s^{(1)} \cdots l_s^{(d)}}_{\text{rank} \le 1}$$

explicit Q

s.t. $rk(M_Q) \ge n^{d/2} \Rightarrow s \ge n^{d/2}$

Example: Tensor Rank

$\{x_1^{(1)}, \dots, x_n^{(1)}\}, \{x_1^{(2)}, \dots, x_n^{(2)}\}, \dots, \{x_1^{(d)}, \dots, x_n^{(d)}\}$

$Q(\bar{x}^{(1)}, \dots, \bar{x}^{(d)}) = l_1^{(1)} \cdots l_1^{(d)} + \cdots + l_s^{(1)} \cdots l_s^{(d)}$

$l_s^{(1)} \cdots l_s^{(d)}$: rank ≤ 1

explicit Q

s.t. $rk(M_Q) \geq n^{d/2} \Rightarrow s \geq n^{d/2}$

$Q = \langle x_1, x_{d/2+1} \rangle \langle x_2, x_{d/2+2} \rangle \cdots \langle x_{d/2}, x_d \rangle$

"Product of Inner Products"

Example: Tensor Rank

$\{x_1^{(1)}, \dots, x_n^{(1)}\}, \{x_1^{(2)}, \dots, x_n^{(2)}\}, \dots, \{x_1^{(d)}, \dots, x_n^{(d)}\}$

$Q(\bar{x}^{(1)}, \dots, \bar{x}^{(d)}) = l_1^{(1)} \cdots l_1^{(d)} + \cdots + \underbrace{l_s^{(1)} \cdots l_s^{(d)}}_{\text{rank} \le 1}$

explicit Q

s.t. $rk(M_Q) \ge n^{d/2} \Rightarrow s \ge n^{d/2}$

$Q = \langle x_1, x_{d/2+1} \rangle \langle x_2, x_{d/2+2} \rangle \cdots \langle x_{d/2}, x_d \rangle$

"Product of Inner Products"

$M_Q = I_n \otimes I_n \otimes \cdots$

Example : Tensor Rank

$\{x_1^{(1)}, \dots, x_n^{(1)}\}, \{x_1^{(2)}, \dots, x_n^{(2)}\}, \dots, \{x_1^{(d)}, \dots, x_n^{(d)}\}$

$Q(\bar{x}^{(1)}, \dots, \bar{x}^{(d)}) = \ell_1^{(1)} \cdots \ell_1^{(d)} + \cdots + \underbrace{\ell_s^{(1)} \cdots \ell_s^{(d)}}_{\text{rank} \le 1}$

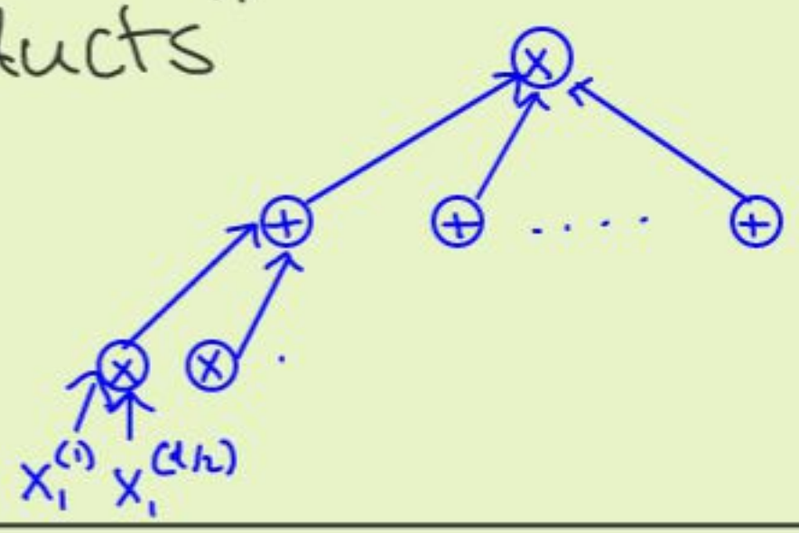
$\Downarrow$

explicit Q s.t. $rk(M_Q) \ge n^{d/2} \Rightarrow s \ge n^{d/2}$

$Q = \langle x_1, x_{d/2+1} \rangle \langle x_2, x_{d/2+2} \rangle \cdots \langle x_{d/2}, x_d \rangle$

"Product of Inner Products"

$M_Q = I_n \otimes I_n \otimes \cdots$



Other applications of Partial derivative method

→ [NW]: homogeneous $\Sigma\Pi\Sigma$ circuits.

Other applications of Partial derivative method

→ [NW]: homogeneous $\Sigma\Pi\Sigma$ circuits.

→ [NW, Raz, RY]: Weaker "subexponential" lower bounds against higher-depth set-multilinear circuits & generalizations.

Other applications of Partial derivative method

→ [NW]: homogeneous $\Sigma\Pi\Sigma$ circuits.

→ [NW, Raz, RY]: Weaker "subexponential" lower bounds against higher-depth set-multilinear circuits & generalizations.

→ Partial derivatives + Random Restrictions.

→ Necessary because Products of Inner Products is $\Sigma\Pi\Sigma\Pi$.

Shifted Partial Derivative Method [GKKS]

→ lower bounds against homogeneous $\Sigma\Pi\Sigma\Pi$ circuits

Shifted Partial Derivative Method [GKKS]

→ lower bounds against homogeneous $\Sigma\Pi\Sigma\Pi$ circuits

→ Partial derivatives of such circuits are "simple"

Shifted Partial Derivative Method [GKKS]

→ lower bounds against homogeneous $\Sigma\Pi\Sigma\Pi$ circuits

→ Partial derivatives of such circuits are "simple"

→ derivative $\times$ monomial of degree $\leq \ell$

$$M_P = \left(\right)$$

Shifted Partial Derivative Method [GKKS]

→ lower bounds against homogeneous $\Sigma\Pi\Sigma\Pi$ circuits

→ Partial derivatives of such circuits are "simple"

→ [FLMS, KLSS, KS]: Lower bounds of $N^{\Omega(\sqrt{d})}$ against homogeneous $\Sigma\Pi\Sigma\Pi$

Shifted Partial Derivative Method [GKKS]

→ Lower bounds against homogeneous $\Sigma\Pi\Sigma\Pi$ circuits

→ Partial derivatives of such circuits are "simple"

→ [FLMS, KLSS, KS]: Lower bounds of $N^{\Omega(\sqrt{d})}$ against homogeneous $\Sigma\Pi\Sigma\Pi$

↓

Asymptotic improvement gives $VP \neq VNP$!

Shifted Partial Derivative Method [GKKS]

→ Lower bounds against homogeneous $\Sigma\Pi\Sigma\Pi$ circuits

→ Partial derivatives of such circuits are "simple"

→ [FLMS, KLSS, KS]: Lower bounds of $N^{\Omega(\sqrt{d})}$ against homogeneous $\Sigma\Pi\Sigma\Pi$

↓

Asymptotic improvement gives $VP \neq VNP$!

Hard polynomial = $IMM_{n,d}$

Lopsided Partial derivatives [LST]

Lopsided Partial derivatives [LST]

→ Product of Inner Prods $(X^{(1)}, \ldots, X^{(d)})$

$$= \left(\sum_{i=1}^{n} X_i^{(1)} X_i^{(d_1+1)}\right)\left(\sum_{i=1}^{n} X_i^{(2)} X_i^{(d_1+2)}\right)\ldots$$

→ Avoiding PIP: variable sets have diff-
-erent sizes!

Lopsided Partial derivatives [LST]

→ Product of Inner Prods $(X^{(1)}, \ldots, X^{(d)})$

$$= \left(\sum_{i=1}^{n} X_i^{(1)} X_i^{(d_1+1)}\right)\left(\sum_{i=1}^{n} X_i^{(2)} X_i^{(d_2+2)}\right)\ldots$$

→ Avoiding PIP: variable sets have diff-erent sizes!

Eg: $|X^{(1)}| = |X^{(2)}| = \cdots = |X^{(t)}| = n$

$|X^{(t+1)}| = \cdots = |X^{(d)}| = n^{\alpha}$

s.t. $|p-q\alpha|$ large for "small" p, q.

Lopsided Partial derivatives [LST]

Thm: Lbd. of $N^{-\Omega(\sqrt{d})}$ against $\Sigma\Pi\Sigma\Pi\Sigma$ set-multilinear circuits.

Lopsided Partial derivatives [LST]

Thm: Lbd. of $N^{-2(\sqrt{d})}$ against $\Sigma\Pi\Sigma\Pi\Sigma$ set-multilinear circuits.

Thm [SW, CKSV]: Depth-structure tradeoff.(*)

$\Sigma\Pi\Sigma$ size $s \Rightarrow$ $\Sigma\Pi\Sigma\Pi\Sigma$ s.m. size $s \cdot d^{O(d)}$

Lopsided Partial derivatives [LST]

Thm: Lbd. of $N^{\Omega(\sqrt{d})}$ against $\Sigma\Pi\Sigma\Pi\Sigma$ set-multilinear circuits.

Thm [SW, CKSV]: Depth-structure tradeoff.(*)

$\Sigma\Pi\Sigma$ size $s \Rightarrow$ $\Sigma\Pi\Sigma\Pi\Sigma$ s.m. size $s \cdot d^{O(d)}$

Cor: $d < \sqrt{\log N} \Rightarrow N^{\Omega(\sqrt{d})}$ lbd. against $\Sigma\Pi\Sigma$

Lopsided Partial derivatives [LST]

Thm: Lbd. of $N^{\Omega(\sqrt{d})}$ against $\Sigma\Pi\Sigma\Pi\Sigma$ set-multilinear circuits.

Thm [SW, CKSV]: Depth-structure tradeoff.(*)

$\Sigma\Pi\Sigma$ size $s \Rightarrow \Sigma\Pi\Sigma\Pi\Sigma$ s.m. size $s \cdot d^{O(d)}$

Cor: $d < \sqrt{\log N} \Rightarrow N^{\Omega(\sqrt{d})}$ lbd. against $\Sigma\Pi\Sigma$

↓

Asymptotic improvement gives $VP \neq VNP$!

Hard polynomial = $IMM_{n,d}$.

Lopsided Partial derivatives [LST]

Thm: Lbd. of $N^{d^{\exp(-\Delta)}}$ against depth Δ set-multilinear circuits.

Thm [LST]: Depth-structure tradeoff$^{(*)}$.

Depth Δ, size s $\Rightarrow$ Depth $2\Delta+1$, s.m. size $s \cdot d^{O(d)}$

Cor: $d < \sqrt{\log N}$ $\Rightarrow$ $N^{d^{\exp(-\Delta)}}$ lbd. against depth Δ

Talk Outline

① Introduction to Algebraic Complexity

② Lower bound Techniques

③ Polynomial Identity Testing

Polynomial Identity Testing

$$(x_1^2+x_2^2+x_3^2+x_4^2)\cdot(y_1^2+y_2^2+y_3^2+y_4^2) = (x_1y_1-x_2y_2-x_3y_3-x_4y_4)^2 + (x_1y_2+x_2y_1+x_3y_4-x_4y_3)^2 + (x_1y_3-x_2y_4+x_3y_1+x_4y_2)^2 + (x_1y_4+x_2y_3-x_3y_2+x_4y_1)^2$$

Polynomial Identity Testing

$$(x_1^2+x_2^2+x_3^2+x_4^2)\cdot(y_1^2+y_2^2+y_3^2+y_4^2) = (x_1y_1-x_2y_2-x_3y_3-x_4y_4)^2 + (x_1y_2+x_2y_1+x_3y_4-x_4y_3)^2 + (x_1y_3-x_2y_4+x_3y_1+x_4y_2)^2 + (x_1y_4+x_2y_3-x_3y_2+x_4y_1)^2$$

Input: $P(x_1,\dots,x_N)$ (as a circuit)

Output: Is $P=0$?

Deterministic Algorithms for PIT & Lower bounds

→ Rand algo : ① Fix $S \subseteq \mathbb{F}$. $|S| \geq 10d$.

② Pick $a \in_R \mathbb{F}^n$. Test if $P(a) = 0$.

Deterministic Algorithms for PIT & Lower bounds

→ Rand algo : ① Fix $S \subseteq \mathbb{F}$. $|S| \geq 10d$.

② Pick $a \in_R \mathbb{F}^n$. Test if $P(a) = 0$.

→ Can we get efficient non-randomized algo?

Deterministic Algorithms for PIT & Lower bounds

→ Rand algo : ① Fix $S \subseteq \mathbb{F}$. $|S| \geq 10d$.

② Pick $a \in_R \mathbb{F}^n$. Test if $P(a) = 0$.

→ Can we get efficient non-randomized algo?

Thm $\begin{bmatrix} KI, \\ Agr \end{bmatrix}$ PIT algos $\Rightarrow$ Lower bounds

→ PIT algos. for weak circuit classes

→ Hardness - randomness tradeoffs.

Deterministic Algorithms for PIT & Lower bounds

→ Rand algo : ① Fix $S \subseteq \mathbb{F}$. $|S| \geq 10d$.

② Pick $a \in_R \mathbb{F}^n$. Test if $P(a) = 0$.

→ Can we get efficient non-randomized algo?

Thm [KI]: Hardness-randomness tradeoffs.

Deterministic Algorithms for PIT & Lower bounds

→ Rand algo : ① Fix $S \subseteq \mathbb{F}$. $|S| \geq 10d$.

② Pick $a \in_R \mathbb{F}^n$. Test if $P(a) = 0$.

→ Can we get efficient non-randomized algo?

Thm [KI]: Hardness-randomness tradeoffs.

Superpolynomial lbds ⟹ Subexponential time PIT for general ckts

Exponential lbds ⟹ Quasipoly. time PIT

Deterministic Algorithms for PIT & Lower bounds

→ Rand algo : ① Fix $S \subseteq \mathbb{F}$. $|S| \geq 10d$.

② Pick $a \in_R \mathbb{F}^n$. Test if $P(a) = 0$.

→ Can we get efficient non-randomized algo?

Thm [KI]: Hardness-randomness tradeoffs.

Superpolynomial lbds ⟹ Subexponential time PIT for general ckts

Exponential lbds ⟹ Quasipoly. time PIT

[GKSS]: Lower bounds ⟹ Poly-time PIT. for $O(1)$-variable polys

Deterministic Algorithms for PIT & Lower bounds

→ Rand algo : ① Fix $S \subseteq \mathbb{F}$. $|S| \geq 10d$.

② Pick $a \in_R \mathbb{F}^n$. Test if $P(a) = 0$.

→ Can we get efficient non-randomized algo?

Thm [DSY, CKS]: Tradeoffs for const. depth

Deterministic Algorithms for PIT & Lower bounds

→ Rand algo : ① Fix $S \subseteq \mathbb{F}$. $|S| \geq 10d$.

② Pick $a \in_R \mathbb{F}^n$. Test if $P(a) = 0$.

→ Can we get efficient non-randomized algo?

Thm [DSY, CKS]: Tradeoffs for const-depth LOW-DEGREE superpoly-lower bds. against $O(1)$-depth ckts $\Rightarrow$ Subexponential time PIT for $O(1)$-depth ckts.

Deterministic Algorithms for PIT & Lower bounds

→ Rand algo : ① Fix $S \subseteq \mathbb{F}$. $|S| \geq 10d$.

② Pick $a \in_R \mathbb{F}^n$. Test if $P(a) = 0$.

→ Can we get efficient non-randomized algo?

Thm [DSY, CKS]: Tradeoffs for const-depth LOW-DEGREE superpoly-lower bds. against $O(1)$-depth ckts $\Rightarrow$ Subexponential time PIT for $O(1)$-depth ckts.

Cor: Lower bounds for $IMM_{n,\sqrt{\log n}} \Rightarrow$ PIT

Talk Outline

① Introduction to Algebraic Complexity

② Lower bound Techniques

③ Polynomial Identity Testing

④ Proof Complexity

Ideal Proof system (IPS) [GP]

Ideal Proof system (IPS) [GP]

→ Based on Hilbert's Nullstellensatz.

$f_1(\bar{x}) = \dots$

$= f_m(\bar{x}) = 0 \implies \exists g_1, \dots, g_m \quad \sum_{i=1}^{m} f_i \cdot g_i = 1$

has no solutions

Ideal Proof system (IPS) [GP]

→ Based on Hilbert's Nullstellensatz.

$f_1(\bar{x}) = \dots$

$= f_m(\bar{x}) = 0 \implies \exists g_1, \dots, g_m \quad \sum_{i=1}^{m} f_i \cdot g_i = 1$

has no solutions

→ IPS: g_i's encoded by alg. ckts.

Ideal Proof system (IPS) [GP]

→ Based on Hilbert's Nullstellensatz.

$f_1(\bar{x}) = \dots$

$= f_m(\bar{x}) = 0 \implies \exists g_1, \dots, g_m \quad \sum_{i=1}^{m} f_i \cdot g_i = 1$

has no solutions

→ IPS: g_i's encoded by alg. ckts.

→ Lbds for IPS $\iff$ VP $\neq$ VNP [ST]

→ Lbds for $O(1)$ depth, char $p \implies AC^0[p]$-Freyc lbds

Ideal Proof system (IPS) [GP]

→ Based on Hilbert's Nullstellensatz.

$f_1(\bar{x}) = \dots$

$= f_m(\bar{x}) = 0 \implies \exists g_1, \dots, g_m \quad \sum_{i=1}^{m} f_i \cdot g_i = 1$

has no solutions

→ IPS: g_i's encoded by alg. ckts.

→ Lbds for IPS $\iff$ VP $\neq$ VNP [ST]

→ Lbds for $O(1)$ depth, char $p \implies AC^0[p]$-Freyc lbds

→ Lbds for restricted ckts [FSTW]

→ Lbds for $O(1)$-depth over char 0 [AF, GHT]

Talk Outline

1. Introduction to Algebraic Complexity
2. Lower bound Techniques
3. Polynomial Identity Testing
4. Proof Complexity
5. Possible Barriers to Lower Bounds

Barriers to rank-based techniques

→ Rank-based technique

Linear $\mathcal{L}: P \longmapsto M_P$

$$rk(\mathcal{L}(\text{small ckt.})) << \max_P rk(\mathcal{L}(P))$$

Barriers to rank-based techniques

→ Rank-based technique

Linear $\mathcal{L}: P \longmapsto M_P$

$rk(\mathcal{L}(\text{small ckt.})) << \max_P rk(\mathcal{L}(P))$

Thm: [EGOW] No rank-based technique can prove $\gg n^{d/2}$ - lbds on tensor rank.

[MGOW]: Extensions to higher notions of rank.

Algebraic Natural Proofs [FSV, GKSS]

Algebraic Natural Proofs [FSV, GKSS]

→ Polynomial $P(x_1, \ldots, x_N)$ = vector of coeffs v_P

→ Natural lower bound = Efficiently-computable polynomial P vanishing on all "easy" v_P.

Algebraic Natural Proofs [FSV, GKSS]

→ Polynomial $P(x_1, \ldots, x_N)$ = vector of coeffs v_P

→ Natural lower bound = Efficiently-computable polynomial P vanishing on all "easy" v_P.

→ All rank-based lower bounds are natural.

Algebraic Natural Proofs [FSV, GKSS]

→ Polynomial $P(x_1,\dots,x_N)$ = vector of coeffs $\vec{v}_P$

→ Natural lower bound = Efficiently-computable polynomial P vanishing on all "easy" $\vec{v}_P$.

→ All rank-based lower bounds are natural.

→ Succinct hitting sets ⇒ no natural lower bound against VP

→ Evidence for [FSV, BIJL] & against [GKRST]

Talk Outline

1. Introduction to Algebraic Complexity
2. Lower bound Techniques
3. Polynomial Identity Testing
4. Proof Complexity
5. Possible Barriers to Lower Bounds

What I didn't cover

→ Monotone Algebraic Circuits

→ GCT & Border Complexity

→ Symmetric Algebraic Circuits

→ Factorizations of Algebraic Circuits.

→ Reconstruction

⋮

Open Questions

→ Stronger lower bounds.

→ Lower bounds for depths $\gg \log \log_5 d$

→ char > 0 [GKKS, CICS, GKSS, LST, ...]

→ Lower bounds for syntactic simpli-
-fication

→ Hierarchy theorems

→ Optimal hardness-randomness tradeoffs

→ Other meta-complexity questions