Crash course on Algebraic Complexity

Amir Shpilka
Tel Aviv University

Rough Plan

Lecture 1: Models of computation, Complexity Classes, Reductions and Completeness, Connection to Boolean world, Structural Results

Lecture 2: Lower Bounds, Partial Derivative Method, Shifted Partial Derivatives

Lecture 3: Polynomial Identity Testing, Hardness-Randomness tradeoffs

Lecture 4: Limitations, Future Directions

The Basics

Plan

- Introduction:
  - Basic definitions
  - Motivation
- Valiant's work:
  - VP, VNP
  - Reductions
  - Completeness

Why consider Algebraic Complexity

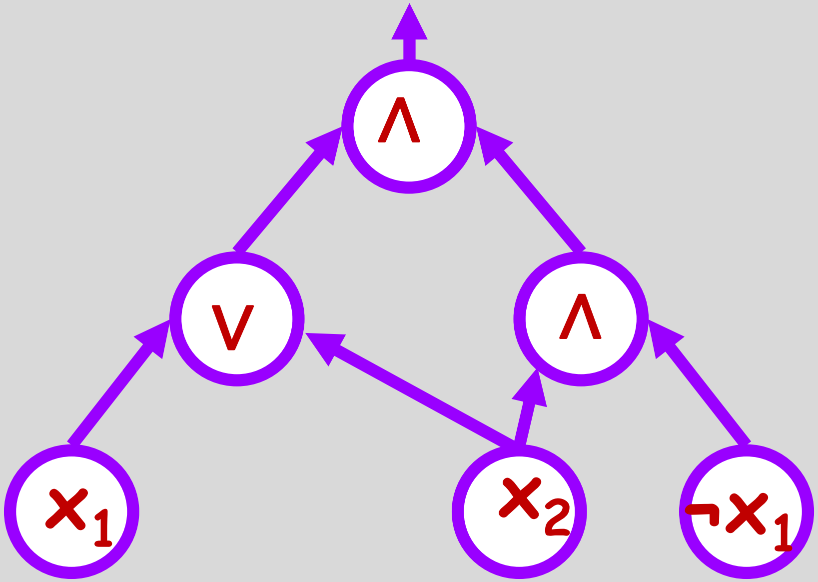
Natural problems are algebraic:

- Linear algebra:
  - Solving a linear system of equations
  - Computing Determinant
  - FFT
- Polynomial Factorization
  - List decoding of Reed-Solomon codes
- Usually computed using Arithmetic Circuits
  - input treated as field elements, basic arithmetic operations at unit cost

Boolean Circuits

Our holy grail: Prove NP $\not\subset$ P/poly

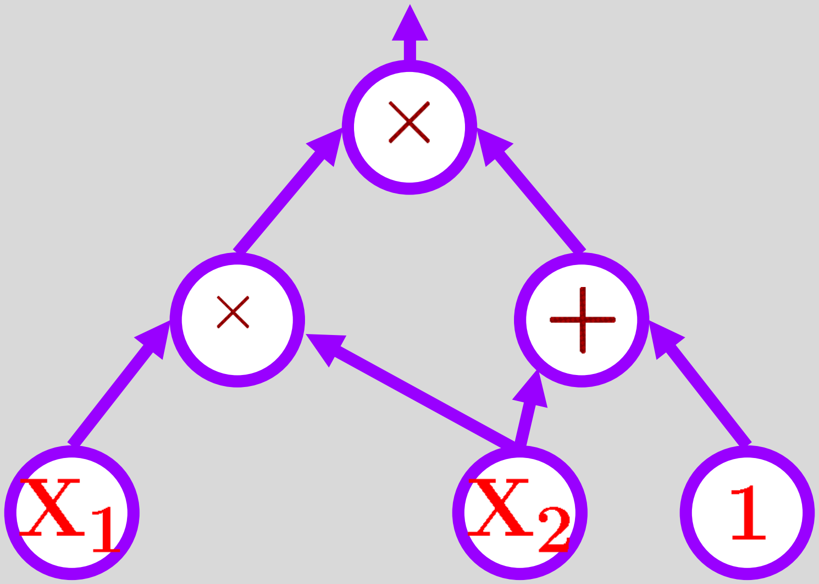
Show that certain problems (e.g., graph-coloring) cannot be decided by small Boolean circuits



Arithmetic Circuits

In Example:

- Size = 6
- Depth = 2
- Degree = 3



Example: $(x_1 \cdot x_2) \cdot (x_2 + 1)$

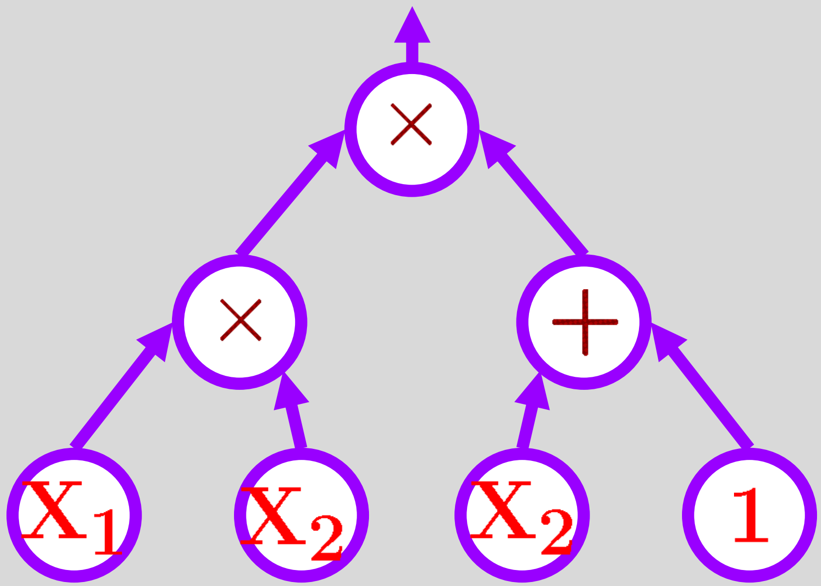
Size = number of wires

Depth = length of longest input-output path

Degree = max degree of internal gates

Arithmetic Formulas

Same, except underlying graph is a tree



Bounded depth circuits

$\Sigma\Pi$ circuits: depth-2 circuits with $+$ at the top and $\times$ at the bottom. Size s circuits compute s-sparse polynomials

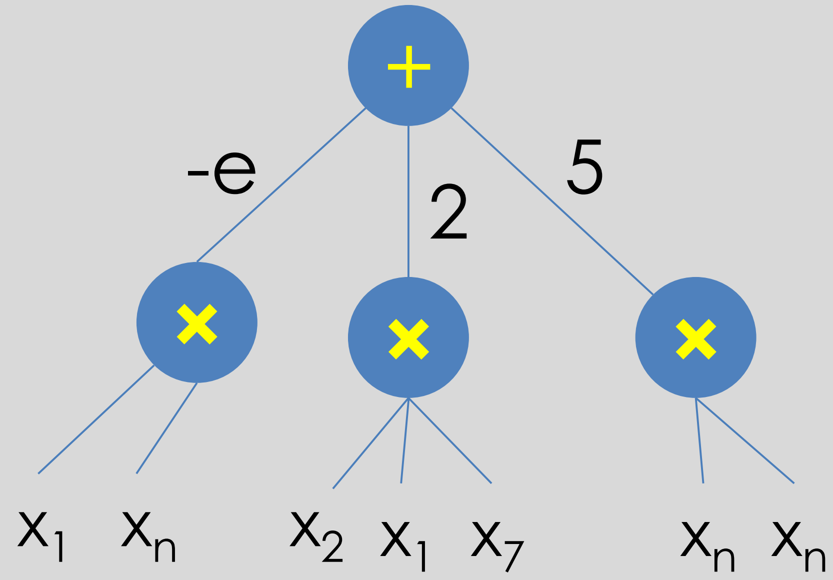
$\Sigma\Pi\Sigma$ circuits: depth-3 circuits with $+$ at the top, $\times$ at the middle and $+$ at the bottom. Compute sums of products of linear functions. I.e. a sparse polynomial composed with a linear transformation

$\Sigma\Pi\Sigma\Pi$ circuits: depth-4 circuits.
Compute sums of products of sparse polynomials

ΣΠ circuits

ΣΠ circuits: depth-2 circuits with + at the top and × at the bottom. Size s circuits compute s-sparse polynomials

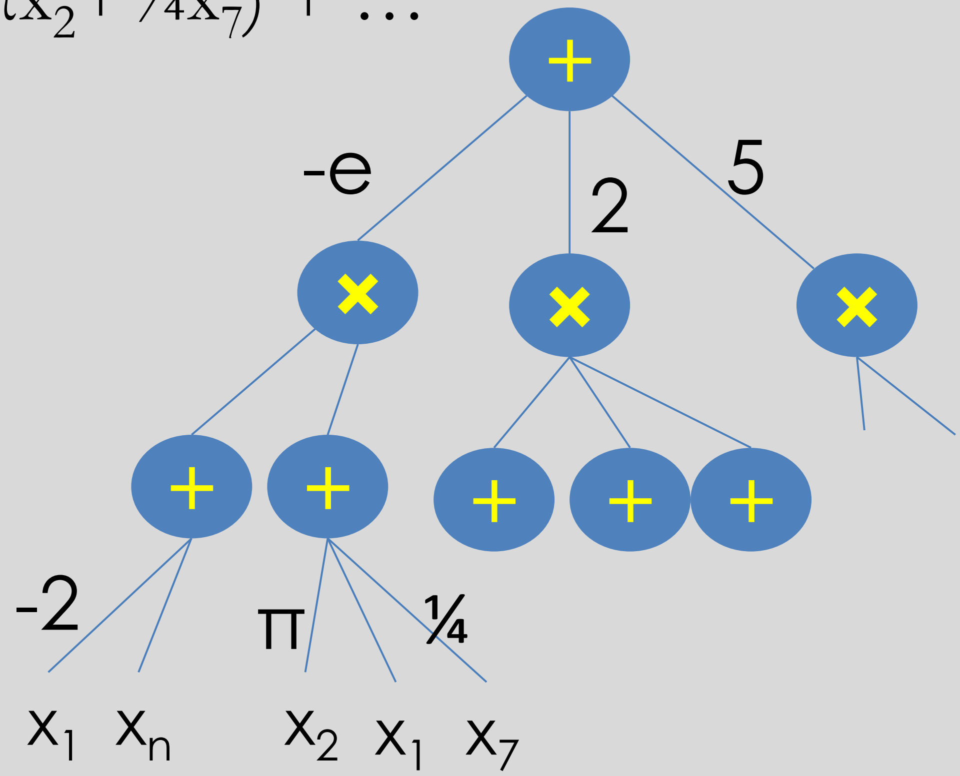
Example: $(-e)x_1 \cdot x_n + 2x_1 \cdot x_2 \cdot x_7 + 5(x_n)^2$



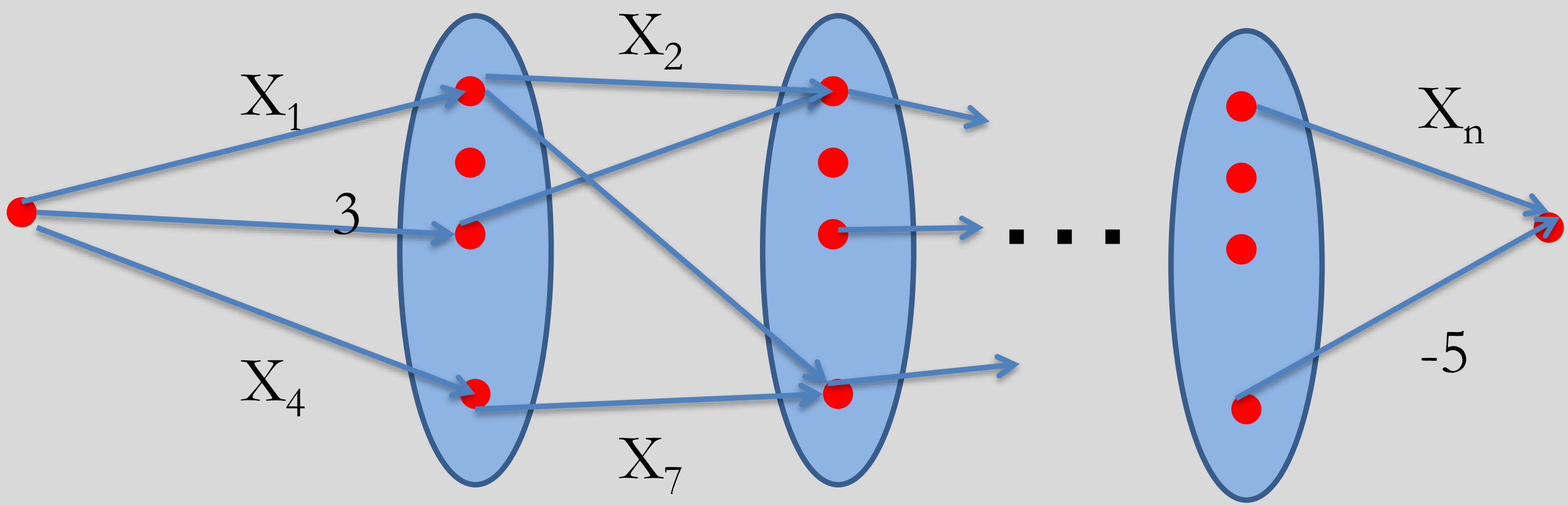
ΣΠΣ circuits

ΣΠΣ circuits: + at the top, × at the middle and + at the bottom: compute sums of products of linear functions

Example: $(-e)\cdot(-2x_1+x_n)\cdot(x_1+\pi x_2+{}^1\!/\!_4 x_7) + \ldots$



Algebraic Branching Programs



Edges labeled by constants/variables

Path computes product of labels

ABP computes sum over paths = product of labeled transition matrices (as in graph powering)

Basic Relations

“Theorem”: Formula $\leq$ ABP $\leq$ Circuits $\leq$ quasi-poly Formula

Basic Relations

“Theorem”: Formula ≤ ABP ≤ Circuits ≤ quasi-poly Formula

Theorem: if f computed by a size s formula then f is computed by an ABP with s edges

Basic Relations

"Theorem": Formula $\leq$ ABP $\leq$ Circuits $\leq$ quasi-poly Formula

Theorem: if f computed by a size s formula then f is computed by an ABP with s edges

Theorem: If f is computed by an ABP with s edges then f computed by an arithmetic circuits of size O(s).

Basic Relations

“Theorem”: Formula $\leq$ ABP $\leq$ Circuits $\leq$ quasi-poly Formula

Theorem: if f computed by a size s formula then f is computed by an ABP with s edges

Theorem: If f is computed by an ABP with s edges then f computed by an arithmetic circuits of size O(s).

Proof: By induction on structure (both cases).

Basic Relations

“Theorem”: Formula $\leq$ ABP $\leq$ Circuits $\leq$ quasi-poly Formula

Theorem: if f computed by a size s formula then f is computed by an ABP with s edges

Theorem: If f is computed by an ABP with s edges then f computed by an arithmetic circuits of size O(s).

Proof: By induction on structure (both cases).

Theorem: “Circuits can be made shallow” i.e. $VP=VNC^2$ (more on that later)

Arithmetic vs. Boolean circuits

Boolean circuits compute Boolean functions: $x = x \wedge x = x \vee x$

Arithmetic circuits compute syntactic objects:

$$x \neq x^2 \text{ as polynomials, even over } \mathbb{F}_2$$

Note: if $\mathbb{F}$ infinite then f=g as polynomials iff f=g as functions

Convention: We only consider families $\{f_n\}$ s.t. $\deg(f_n)=\mathrm{poly}(n)$

- In the Boolean world every function is a multilinear polynomial
- For circuits and inputs with polynomial bit complexity output is also of polynomial bit complexity

Why Arithmetic Circuits?

- Most natural model for computing polynomials
- For many problems (e.g. Matrix Multiplication, DFT) best algorithm is an arithmetic circuit
- Great algorithmic achievements:
  - Fourier Transform
  - Matrix Multiplication
  - Polynomial Factorization
- Structured model (compared to Boolean circuits) P vs. NP may be easier (also true in a formal way)
- Personal view: offers the most natural approach to P vs. NP

Important Problems

- Designing new algorithms:
  - $\tilde{O}(n^2)$ for Matrix Multiplication?
  - Understanding P
- Proving lower bounds:
  - Find a polynomial (e.g. Permanent) that requires super-polynomial size or super-logarithmic depth
  - Analog of NC vs. #P
- Derandomizing Polynomial Identity Testing:
  - Understanding the power of randomness
  - Analog of P vs. RP, BPP

Plan

- ✓ Introduction:
  - Basic definitions
  - Motivation
- Valiant’s work:
  - VP, VNP
  - Reductions
  - Completeness

Complexity Classes – Valiant's work

Efficient computations: A family $\{f_n\}$ is in VP if there exists a polynomial $s:\mathbb{N} \to \mathbb{N}$ such that

- $\#vars(f_n)$, $deg(f_n) < s(n)$
- f_n computed by size $s(n)$ arithmetic circuit

Example: $\{Det_{nxn}\}$ is in VP

Example: $\{x^{2^n}\}$ is not in VP (but has a small circuit)

Similar definition (except degree bound) to P/poly

Note: accurate definition is $VP_{\mathbb{F}}$ as field may matter

Complexity Classes – VNP

Recall: L={L_n}∈NP if there is R(x,y)∈P such that

$$x \in L_n \Leftrightarrow \vee_y R(x,y) = \text{True}$$

Def: A family {f_n}∈VNP if there is {g_n}∈VP such that

$$f_n(x_1, \dots, x_n) = \sum_{y \in \{0,1\}^t} g_n(x_1, \dots, x_n, y_1, \dots, y_t)$$

where t is polynomial in n

Example: Perm(X)= $\sum_\sigma \prod_i x_{i,\sigma(i)} \in$ VNP

$$Perm(X) = \Sigma_{y \in \{0,1\}^n} \Pi_i (2y_i - 1) \Pi_j (x_{j,1} y_1 + \cdots + x_{j,n} y_n)$$

Thumb rule: $f = \Sigma_e c_e \Pi_i x_i^{e_i}$ in VNP if c_e efficiently computable given e

Completeness and Reductions

Reductions: $\{f_n\}$ reduces to $\{g_n\}$ if for some polynomial $t(n)$

$$f_n(x_1,\ldots,x_n) = g_{t(n)}(y_1,\ldots,y_{t(n)})$$

where $y_i \in \{x_1,\ldots,x_n,\} \cup \mathbb{F}$.

I.e., we substitute variables and field elements to the variables of g and get f (also called projection)

Theorem [Valiant]: Perm is complete for VNP (except over characteristic 2)

Theorem [Mahajan-Vinay]: Det is complete for "ABPs"

Valiant's hypothesis: VP $\neq$ VNP

Extended hypothesis: Perm is not a projection of $\mathrm{Det}_{\text{quasi-poly}}$

Theorem [Mignon-Ressayre, Cai-Chen-Li]:

If $\mathrm{Det}(A) = \mathrm{Perm}(X)$ then $\dim(A) = \Omega(n^2)$

Cook's versus Valiant's Hypothesis

Theorem [Valiant]: 0/1 Perm is complete for #P

Building on $PH \subseteq P^{\#P}$ and $VP=VNC^2$ we get

Theorem [Ibarra-Moran, von zur Gathen, Bürgisser]:

- If $VP=VNP$ over $\mathbb{C}$ then (under GRH) $NC^3/poly = P/poly = NP/poly = PH/poly$
- If $VP=VNP$ over $\mathbb{F}_p$ then $NC^2/poly = P/poly = NP/poly = PH/poly$

And, in either cases, $PH=\Sigma_2$

My take: $NP \not\subseteq P/poly$ implies $VP \neq VNP$ so we better start with the Algebraic world

Summary - introduction

- Models: Formula $\leq$ ABP $\leq$ Circuits $\leq$ quasi-poly Formula. Also saw $\Sigma\Pi$, $\Sigma\Pi\Sigma$ circuits
- Complexity Classes: VP, VNP
- Reductions and Completeness: IMM, Det for ABPs, Perm for VNP
- Valiant's hypothesis: Perm does not have poly size circuits
- Extended hypothesis: Perm is not a projection of a quasi-poly-sized determinant

Structural Results

Plan

- Homogenization
- Divisions?
- Depth Reduction
  - VP=VNC^2
  - Reduction to depth 4
- Baur Strassen theorem (computing first order partial derivatives)

Homogenization

Def: f is homogeneous if all monomials have same total degree (e.g., Det. Perm)

Def: Formula/ABP/Circuit is homogeneous if every gate computes a homogeneous polynomial

Theorem (Homogenization): f of degree r has size s circuit(ABP) then f has size $O(r^2 s)$ homogeneous circuit (ABP) computing its homogeneous components

Proof idea: Split every gate to r+1 gates where k'th copy computes homogeneous part of degree k

Open: Homogenizing formulas efficiently (known for degree O(log s) [Raz])

Divisions

Getting rid of divisions [Strassen]: If degree-r f computed in size-s using divisions then f computed by poly(r,s)-size with no divisions

Proof idea:

- transform circuit to one with a single division gate at top (by splitting each gate to numerator and denominator)
- w.l.og. (by translating variables and rescaling) $f = g/(1-h)$ where h has no free term
- $f=g(1+h+h^2+\ldots+h^r+\ldots)$ can stop after h^r and then compute relevant homogeneous parts

Depth Reduction

Theorem (Balancing formulas): f has size s formula then f has depth O(log s) formula

Proof idea: Similar to balancing trees or Boolean formulas

Theorem [Valiant-Skyum-Berkowitz-Rackoff]: $VP=VNC^2$. Any size s, deg r circuit can be transformed to a size poly(s,r), deg r, depth $\log(s)\cdot\log(r)$ circuit

(very rough) Proof idea: use induction to write each gate as

$$f_v = \sum_{i=1}^{s} g_{i1}\cdot g_{i2}\cdot g_{i3}\cdot g_{i4}\cdot g_{i5},$$

where $\deg(g_{ij}) \leq r/2$, and $\{g_{ij}\}$ computed in poly(s)-size

Depth Reduction – all the way down

Theorem: [Agrawal-Vinay, Gupta-Kamath-Kayal-Saptharishi]: Homogeneous f of degree r has size s circuits then

- f has homogeneous $\Sigma\Pi\Sigma\Pi^{[\sqrt{r}]}$ circuit of size $s^{O(\sqrt{r})}$
- (over $\mathbb{C}$) f has depth-3 circuit of size $s^{O(\sqrt{r})}$

Corollary: exponential lower bounds for hom. depth 4 or depth 3 give exponential lower bounds for general circuits

Proof idea: As before each gate is $f_v = \sum_{i=1}^{s} g_{i1} \cdot g_{i2} \cdot g_{i3} \cdot g_{i4} \cdot g_{i5}$ where $\deg(g_{ij}) \leq r/2$. As long as some g_{ij} has degree larger than $\sqrt{r}$ replace it with a similar expression. Process terminates with a $\Sigma\Pi\Sigma\Pi^{[\sqrt{r}]}$ circuit

Baur-Strassen theorem

Theorem [Baur-Strassen]: If f has size s, depth d circuit then $\partial f/\partial x_1 \ldots, \partial f/\partial xn$ have size O(s), depth O(d) circuit.

Proving lower bound for computing n polynomials as hard as proving a lower bound for a single polynomial.

Proof idea: structural induction and derivative rules

Open: What about computing $\{\partial^2 f/\partial x_k \partial x_m\}_{k,m}$?

If in size O(s), then Matrix Multiplication has $O(n^2)$ algorithm (consider $x^t \cdot A \cdot B \cdot y$)

Open: What about computing $\{\partial^2 f/\partial x_k \partial x_k\}_k$?

Summary – structural results

- Homogenization – wlog circuits are homogeneous
- Divisions: no need for those
- $VP=VNC^2$
- Depth reduction: Exponential lower bounds for homogeneous depth 4 circuits imply exponential lower bounds for general circuits
- Baur-Strassen: Computing first order partial derivatives with no extra cost

Lower Bounds

Plan

- Survey of known lower bounds
- Some proofs:
  - General lower bounds
    - Strassen's nlog(n) lower bound
    - n^2 lower bound for ABPs/Formulas
  - Bounded depth circuits
    - Approximation method for $\Sigma\Pi\Sigma$ circuits over $\mathbb{F}_p$
  - Partial derivative method and applications
    - $\Sigma\Pi\Sigma$ circuits
    - Multilinear formulas
  - Shifted partial derivatives method
    - Application for $\Sigma\Pi\Sigma\Pi$ circuits

General lower bounds

Counting arguments (dimension arguments): Most degree n polynomials require exponential sized circuits (even with 0/1 coefficients)

Counting arguments: most linear transformations require $\Omega(n^2)$ operations

Theorem [Strassen]: $\Omega(n \cdot \log r)$ lower bound for computing (simultaneously) $x_1^r, x_2^r, \ldots, x_n^r$

Theorem[Baur–Strassen]: same for $x_1^r + \ldots + x_n^r$

No lower bounds for constant degree polynomials

Theorem: [Kalorkoti, Kumar, Chatterjee-Kumar-She-Volk] $\Omega(nr)$ lower bound for formulas/ABPs

Lower Bounds for Small Depth Circuits
(recall exponential bounds for Boolean $AC^0[p]$)

Depth-2 is trivial (sum of monomials)

Over $\mathbb{F}_2$ [Razborov,Smolensky] classical lower bounds hold

[Grigoriev-Karpinski, Grigorev-Razborov]: exp. lower bounds for $\Sigma\Pi\Sigma$ circuits over $\mathbb{F}_p$ (approximation method)

[Nisan-Wigderson]: exp. lower bounds for homogeneous/low degree $\Sigma\Pi\Sigma$ circuits

[S-Wigderson, Kayal-Saha-Tavenas]: ~~quadratic~~ cubic lower bounds over $\mathbb{Q}, \mathbb{C}$ for $\Sigma\Pi\Sigma$ circuits

Open: strong lower bounds for depth-3 circuits over $\mathbb{Q}, \mathbb{C}$

Recall: by [Gupta-Kamath-Kayal-Saptharishi] exponential lower bounds for depth-3 may be hard…

Lower Bounds for Small Depth Circuits (recall exponential bounds for Boolean $AC^0[p]$)

Recall: [Agrawal-Vinay, Gupta-Kamath-Kayal-Saptharishi]: f has size s homogeneous circuit then f has $\Sigma\Pi\Sigma\Pi^{[\sqrt{r}]}$ homogeneous circuit of size $s^{O(\sqrt{r})}$

[Gupta-Kamath-Kayal-Saptharishi, …]: $s^{\Omega(\sqrt{r})}$ lower bounds for homogeneous $\Sigma\Pi\Sigma\Pi^{[\sqrt{r}]}$ circuits

Lower bounds fall short of implying lower bound for general circuit (constant in exponent too small!)

Even "worse" [Fourier-Limaye-Malod-Srinivasan,Kumar-Saraf]: lower bounds hold for easy polynomials, e.g., IMM

[Raz]: $n^{1+O(1/d)}$ lower bound for depth d circuits

Multilinear Models

Gates compute multilinear/homogeneous polynomials

[Raz]: DET,PERM require quasi-poly mult. formulas

$$\text{mult-NC}^1 \subsetneq \text{mult-NC}^2$$

[Raz-Yehudayoff]: $\exp(n^{\Omega(1/d)})$ bounds for depth d multilinear circuits

[Raz-S-Yehudayoff, Alon-Kumar-Volk]: n^2 lower bound for multilinear circuits

Plan

✓ Survey of known lower bounds

- Some proofs:
  - General lower bounds
    - Strassen's nlog(n) lower bound
    - n^2 lower bound for ABPs/Formulas
  - Bounded depth circuits
    - Approximation method for $\Sigma\Pi\Sigma$ circuits over $\mathbb{F}_p$
  - Partial derivative method and applications
    - $\Sigma\Pi\Sigma$ circuits
    - Multilinear formulas
  - Shifted partial derivatives method
    - Application for $\Sigma\Pi\Sigma\Pi$ circuits

Strassen's lower bound

Recall: $\Omega(n \cdot \log r)$ lower bound for $x_1^r, x_2^r, \ldots, x_n^r$

Bézout's Theorem: $f_1,\ldots, f_k$ polynomials in $x_1,\ldots,x_n$ of degrees $r_1,\ldots, r_k$. For every $b_1,\ldots, b_k$ in $\mathbb{F}$ the number of solutions to $f_1(x_1,\ldots,x_n) = b_1,\ldots, f_k(x_1,\ldots,x_n) = b_k$ is infinite or at most $r_1 \cdot \ldots \cdot r_k$

Example: $f_i = x_i^r$, $b_i = 1$, $i=1,\ldots,n$.
The number of solutions is r^n over $\mathbb{C}$

Strassen's lower bound

Assume a circuit of size s for $x_1^r, x_2^r, \ldots, x_n^r$

Associate a variable y_v with every gate v

For each gate $v = u \text{ op } w$ set an equation $y_v - (y_u \text{ op } y_w) = 0$

For an input v set $y_v - x_v = 0$

For an output v set, in addition, $y_v = 1$

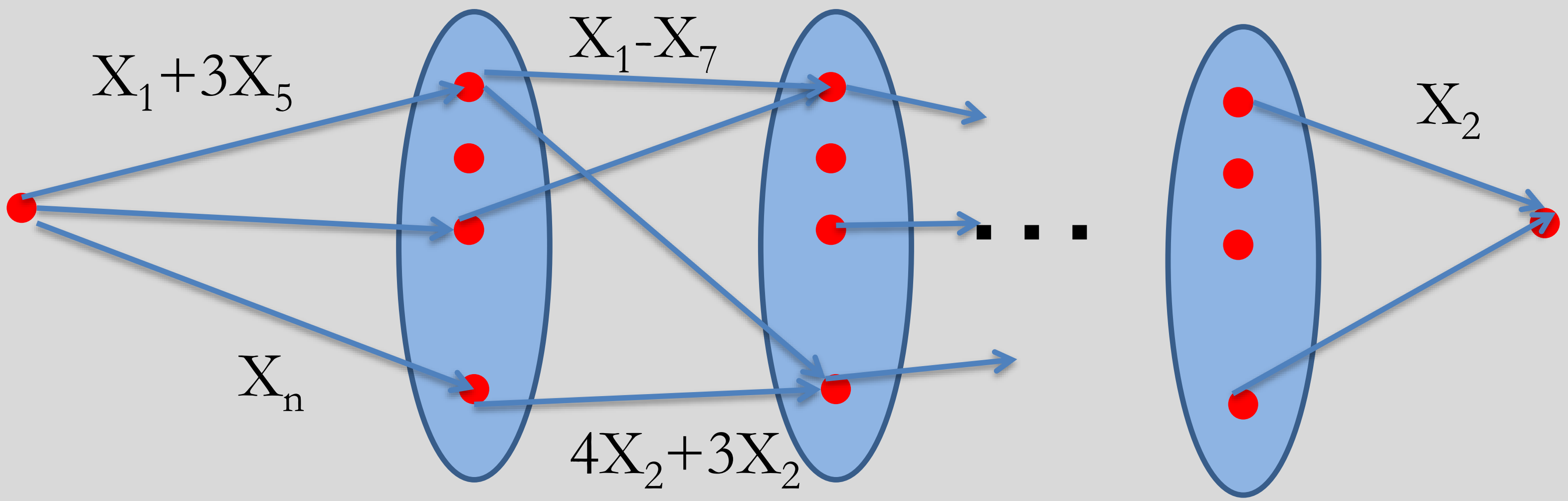
Any solution (in x,y) to the system gives a solution to $\{x_i^r = 1\}$ and vice versa.

By Bézout at most 2^s solutions (finite number of solutions and s equations of degree at most 2 each)

Hence $2^s \geq r^n$ (can replace s by # of multiplications)

Note: cannot get bound better than $n \cdot \log r$

Kumar's lower bound for homogeneous ABPs



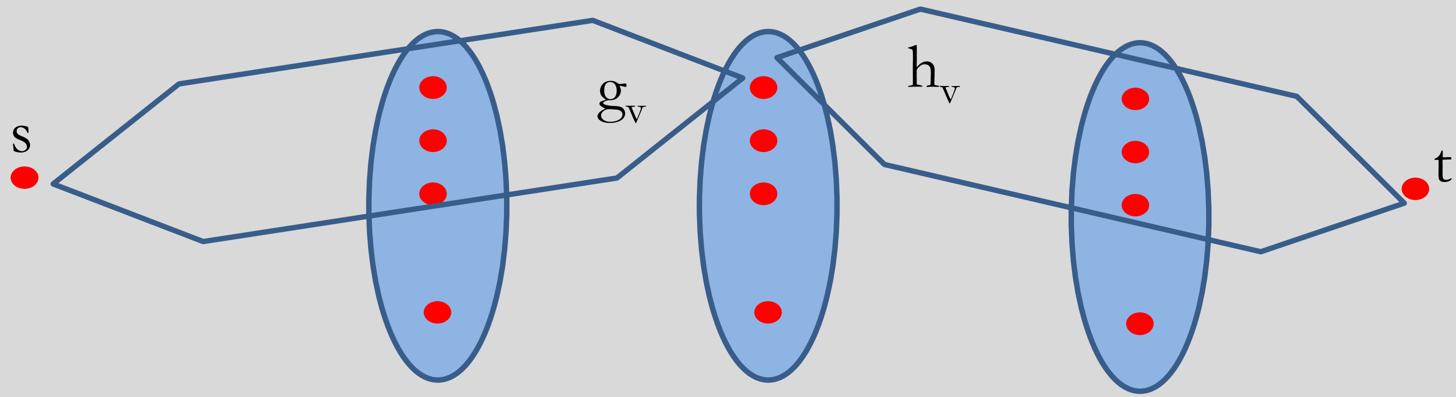
Recall: ABP computes sum (over paths) of products of labels on path

Edges labeled by linear forms

Homogeneous ABP: vertices compute homogeneous polys

Note: Vertices in level j compute degree j polynomials

Kumar's lower bound for homogeneous ABPs



g_v computed by [s,v] and h_v by [v,t] (v in layer j, L_j)

Then, $f = \sum_{v \; in \; L_j} g_v \cdot h_v$

Main Lemma: if $x_1^r + x_2^r + \cdots x_n^r = \sum_{i=1}^m g_i \cdot h_i$ all are homogeneous and non constant then $m \geq n/2$

Proof idea: Common zero of $\{g_i, h_i\}$ is a zero of $(x_1^{r-1}, \ldots, x_n^{r-1})$. Only one zero so result follows by dimension arguments

Note: n/2 lower bound also for Determinantal complexity

Plan

- ✓ Survey of known lower bounds
- Some proofs:
  - ✓ General lower bounds
    - ✓ Strassen's nlog(n) lower bound
    - ✓ n^2 lower bound for ABPs/Formulas
  - – Bounded depth circuits
    - Approximation method for $\Sigma\Pi\Sigma$ circuits over $\mathbb{F}_p$
  - – Partial derivative method and applications
    - $\Sigma\Pi\Sigma$ circuits
    - Multilinear formulas
  - – Shifted partial derivatives method
    - Application for $\Sigma\Pi\Sigma\Pi$ circuits

Approximation method for ΣΠΣ circuits

[Grigoriev-Karpinski, Grigoriev-Razborov]: lower bounds over $\mathbb{F}_p$ (a-la Razborov-Smolensky for $AC^0[p]$ circuits):

- If a multiplication gate contains $n^{1/2}$ linearly independent functions then it is 0, except with probability $\exp(-n^{1/2})$
- A function in k linear functions has degree $< pk$
- Hence, a circuit with s multiplication gates computes a polynomial that is $s \cdot \exp(- n^{1/2})$ close to a degree $O(n^{1/2})$ polynomial
- Correlation bounds for $\mathrm{Mod}(q)$ give $\exp(n^{1/2})$ lower bound

Question: But what about char 0?

Plan

- ✓ Survey of known lower bounds
- Some proofs:
  - ✓ General lower bounds
    - ✓ Strassen's nlog(n) lower bound
    - ✓ n^2 lower bound for ABPs/Formulas
  - ✓ Approximation method for $\Sigma\Pi\Sigma$ circuits over $\mathbb{F}_p$
  - – Partial derivative method and applications
    - $\Sigma\Pi\Sigma$ circuits
    - Multilinear formulas
  - – Shifted partial derivatives method
    - Application for $\Sigma\Pi\Sigma\Pi$ circuits

Partial Derivative Method [Nisan]

[Nisan-Wigderson] exponential lower bounds for homogeneous (or low degree) depth 3 circuits

[S-Wigderson] n^2 lower bound for depth 3 circuits

[Raz]: Det,Perm require quasi-poly multilinear Formulas

[Raz]: multilinear-$NC^1 \subsetneq$ multilinar-NC^2

[Raz-Yehudayoff]: $\exp(n^{\Omega(1/d)})$ bounds for depth d multilinear Circuits

[Raz-S-Yehudayoff, Alon-Kumar-Volk]: n^2 lower bound for multilinear circuits

Partial Derivatives as Complexity Measure

Def: $\partial^{=k}(f) = \{\partial^k f/\partial x_{i_1}\partial x_{i_2}\ldots\partial x_{i_k}\}$ = set of all partial derivatives of f of order k.

Def: $\mu_k(f) = \dim(\mathrm{span}(\partial^{=k}(f))$

In words, take all partial derivatives of order k of f and compute the dimension of their span

Intuition: not easy to create "uncorrelated" partial derivatives

Example: $f = \mathrm{Det}(X)$

$\partial^{=k}(f) = \{\mathrm{Det}(X_{I,J}) : |I| = |J| = n\text{-}k\}$

$\mu_k(f) = \dim(\mathrm{span}(\partial^{=k}(f)) = \binom{n}{k}^2$

Basic Properties of Partial Derivatives

Recall: $\mu_k(f) = \dim(\mathrm{span}(\partial^{=k}(f))$

Basic properties:

- $\mu_k(f + g) \leq \mu_k(f) + \mu_k(g)$
- $\mu_k(f \cdot g) \leq \sum_t \mu_t(f) \cdot \mu_{k-t}(g)$
- $\mu_k(\ell^r) \leq 1$ $(\partial^k \ell^r / \partial x i_1 \partial x i_2 \ldots \partial x ik = c \cdot \ell^{r-k})$
- $\mu_k(\prod_{i=1}^r \ell_i) \leq \binom{r}{k}$ (spanned by all products of r-k of the linear functions)

Lower Bounds for $\Sigma\Lambda\Sigma$ circuits

$\Sigma\Lambda\Sigma$ circuits compute polynomials of the form

$$f = \sum_{i=1}^{s} \ell_i^r$$

Claim: $\mu_k(f) \le s$

Proof: $\mu_k(\ell^r) \le 1$ and subadditivity.

Corollary: Any $\Sigma\Lambda\Sigma$ circuit computing $x_1 \cdot x_2 \cdots x_n$ has size $\exp(\Omega(n))$

Lower Bounds for homogeneous ΣΠΣ circuits

Homogeneous ΣΠΣ circuits compute polynomials of the form

$$f = \sum_{i=1}^{s} \prod_{j=1}^{r} \ell_{i,j}$$

Claim: $\mu_k(f) \leq s \cdot \binom{r}{k}$

Proof: $\mu_k(\prod_{i=1}^{r} \ell_i) \leq \binom{r}{k}$ and subadditivity

Corollary [Nisan-Wigderson]: Any homogeneous ΣΠΣ circuit computing Det/Perm has size $\exp(\Omega(n))$

Lower Bounds for $\Sigma\Pi\Sigma$ circuits

Let $\sigma_n^r(x) = \sum_{|T|=r} \prod_{i \in T} x_i$

Theorem [S-Wigderson]: $\Sigma\Pi\Sigma$ size of $\sigma_n^{\log(n)}(x)$ is $\tilde{\Omega}(n^2)$

Proof: If more than n/10 multiplication gates of degree at least n/10 then we are done. Otherwise, there exists a subspace V of dimension 0.9n such that restricted to V, $\sigma_n^{\log(n)}(x)$ has small circuit of degree at most n/10.

Claim: $\mu_r\left(\sigma_n^{2r}(x)|_V\right) \geq \binom{0.9n}{r}$

Claim: $\mu_r(\sum \prod \sum |_V) \leq \binom{n/10}{r}$

Upper Bounds for $\Sigma\Pi\Sigma$ circuits

Theorem [Ben-Or]: $\Sigma\Pi\Sigma$ size of $\sigma_n^r(x)$ is $O(n^2)$

Proof: Evaluate $f(y)=(y+x_1)\ldots(y+x_n)$ at $n+1$ points, then take the appropriate linear combination to get the coefficient of y^{n-r} which is $\sigma_n^r(x)$

Submodel of $\Sigma\Pi\Sigma$ circuits [S]: $f = \sigma_s^r(\ell_1, \ldots, \ell_s)$ f is a restriction of $\sigma_s^r(x)$ to an n dimensional subspace (can compute any f like that)

[Kayal-Saha-Tavens]: $\widetilde{\Omega}\,(n^2)$ lower bound for an explicit multilinear polynomial in VNP

Open: Prove super quadratic lower bounds

Upper Bounds for ΣΠΣ circuits

Recall [Ryser]: $\mathrm{Perm}(X)$
$$= \Sigma_{y \in \{0,1\}^n} \Pi_i (2y_i - 1) \Pi_j (x_{j,1} y_1 + \cdots + x_{j,n} y_n)$$

This is a ΣΠΣ circuit of size exp(n). What about Det?

Recall [Gupta-Kamath-Kayal-Saptharishi]: f has size s circuits (over $\mathbb{C}$) then f has ΣΠΣ circuit of size $s^{O(\sqrt{r})}$

Corollary: Det has ΣΠΣ complexity $\exp(\tilde{O}(\sqrt{n}))$

Only known construction via [GKKS].

Open: A "nice" ΣΠΣ circuit for Det

Plan

- ✓ Survey of known lower bounds
- Some proofs:
  - ✓ General lower bounds
    - ✓ Strassen's nlog(n) lower bound
    - ✓ n^2 lower bound for ABPs/Formulas
  - ✓ Approximation method for ΣΠΣ circuits over $\mathbb{F}_p$
  - – Partial derivative method and applications
    - ✓ ΣΠΣ circuits
    - Multilinear formulas
  - – Shifted partial derivatives method
    - Application for ΣΠΣΠ circuits

Partial Derivative Matrix [Nisan]

f a multilinear polynomial over $\{y_1,\ldots,y_m\} \sqcup \{z_1,\ldots,z_m\}$

Def: $M_f = 2^m$ dimensional matrix:

Rows indexed by multilinear monomials in $\{y_1,\ldots,y_m\}$

Columns indexed by multilinear monomials in $\{z_1,\ldots,z_m\}$

$M_f(p,q)$ = coefficient of $p \cdot q$ in f

$\mu_{y|z}(f) = \mathrm{rank}(M_f)$

Note: $\mu_{y|z}(f) \leq 2^m$

Def: f is full rank if $\mu_{y|z}(f) = 2^m$

Examples

$f(y,z) = 1+ay+bz+abyz$

$\mu_{y|z}(f) = 1$

$$M_f = \begin{array}{cc} \mathbf{1} & \mathbf{z} \\ \end{array}$$

$$M_f = \begin{array}{c|c} \begin{matrix} \mathbf{1} & \mathbf{z} \end{matrix} & \\ \begin{bmatrix} \mathbf{1} & b \\ a & ab \end{bmatrix} & \begin{matrix} \mathbf{1} \\ \mathbf{Y} \end{matrix} \end{array}$$

$f(y_1,y_2,z_1,z_2) =$

$1 + y_1y_2 - y_1z_1z_2$

$\mu_{y|z}(f) = 2$

$$M_f = \begin{array}{cc} \begin{matrix} \mathbf{1} & \mathbf{z_1} & \mathbf{z_2} & \mathbf{z_1z_2} \end{matrix} & \\ \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{-1} \\ 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 \end{bmatrix} & \begin{matrix} \mathbf{1} \\ \mathbf{y_1} \\ \mathbf{y_2} \\ \mathbf{y_1y_2} \end{matrix} \end{array}$$

Basic facts for a multilinear f

- If f depends on only k variables in $\{y_1,\dots,y_m\}$ then $\mu_{y|z}(g) \le 2^k$
- If $f = g + h$ then $\mu_{y|z}(f) \le \mu_{y|z}(g) + \mu_{y|z}(h)$
- If $f = g \cdot h$ then $\mu_{y|z}(f) = \mu_{y|z}(g) \cdot \mu_{y|z}(h)$
- Corollary: If $f = L_1 \cdot L_2 \cdot \dots \cdot L_k$ = product of linear functions then $\mu_{y|z}(f) \le 2^k$

Unbalanced Gates

Y_f = variables in $\{y_1,\ldots,y_m\}$ that f depends on

Z_f = variables in $\{z_1,\ldots,z_m\}$ that f depends on

Def: f is k-unbalanced if $|\#Y_f - \#Z_f| \geq k$

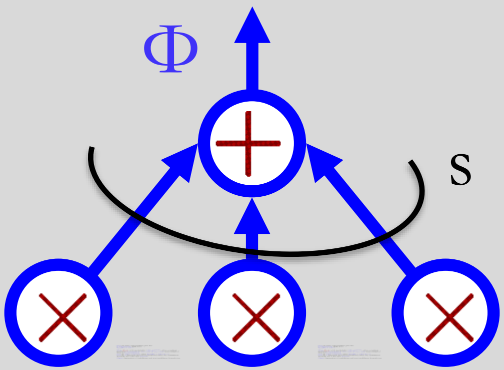
A gate v is k-unbalanced if it computes a k-unbalanced function

Main observation: If $f=g\cdot h$ and either g or h are k-unbalanced then $\mu_{y|z}(f) \leq 2^{m-k}$

Proof: W.l.o.g. $|Y_g|-|Z_g|\geq k$. Hence, $|Z_h|-|Y_h|\geq k$ and

$\mu_{y|z}(f) = \mu_{y|z}(g) \cdot \mu_{y|z}(h) \leq \min(2^{|Zg|}\cdot 2^{|Yh|}, 2^{|Yg|}\cdot 2^{|Zh|}) \leq 2^{m-k}$

Lower bounds for multilinear formulas



Cor: if every top product gate has k-unbalanced child then $\mu_{y|z}(\Phi) \leq s \cdot 2^{m-k}$

Thm [Raz]: with probability $|\Phi| \cdot m^{-\Omega(\log m)}$, after a random partition $\{x_1,\dots,x_{2m}\} = \{y_1,\dots,y_m\} \sqcup \{z_1,\dots,z_m\}$ every child of root is m^{ε}-unbalanced

Cor: If $|\Phi| < m^{O(\log m)}$ then $\mu_{y|z}(\Phi) < |\Phi| \cdot 2^{m - m^{\varepsilon}}$

Cor: If f full rank (for most partitions) then any multilinear formula for f has size $m^{\Omega(\log m)}$

Open: Separation of multilinear and non-multilinear formula size

Limitation of Partial Derivative method

Consider $\Sigma\Lambda\Sigma\Pi^{[2]}$ circuits computing polynomials of the form $Q_1^r+\ldots+Q_s^r$, where each Q_i is quadratic

What is the complexity of the monomial $f=x_1\cdot\ldots\cdot x_n$ in this model? Intuitively, shouldn't be easy to compute

We already saw $\mu_k(f) = \binom{n}{k}$

However, for $g = x_1^2+\ldots+x_n^2$ we have $\mu_k(g)\geq\binom{n}{k}$

Thus, partial derivative method fail to give meaningful bounds even for $\Sigma\Lambda\Sigma\Pi^{[2]}$ circuits

Plan

✓ Survey of known lower bounds

- Some proofs:
  - ✓ General lower bounds
    - ✓ Strassen's nlog(n) lower bound
    - ✓ n^2 lower bound for ABPs/Formulas
  - ✓ Approximation method for $\Sigma\Pi\Sigma$ circuits over $\mathbb{F}_p$
  - ✓ Partial derivative method and applications
    - ✓ $\Sigma\Pi\Sigma$ circuits
    - ✓ Multilinear formulas
  - – Shifted partial derivatives method
    - Application for $\Sigma\Pi\Sigma\Pi$ circuits

Shifted Partial Derivatives

Complexity measure introduced by [Kayal]:

Def: $\mu_k^\ell(f) = \dim(\mathrm{span}(\bar{x}^\ell \cdot \partial^{=k}(f))$

In words, take all partial derivatives of order k of f, multiply each of them by every possible monomial of degree $\leq \ell$ and compute the dimension of the span

Example: $g=x^2$, $f = xy$

- $\bar{x}^1 \cdot \partial^{=1}(g) = \{1,x,y\} \cdot \{x^2\} = \{x^2,x^3,x^2y\}$
- $\bar{x}^1 \cdot \partial^{=1}(f) : \{1,x,y\} \cdot \{x,y\} = \{x,y, x^2,xy, y^2\}$
- $\mu_1^1(g)=3$, $\mu_1^1(f)=5$

Basic properties:

- $\mu_k^\ell(f+g) \le \mu_k^\ell(f) + \mu_k^\ell(g)$
- $\mu_k^\ell(x_1 \cdot \dots \cdot x_n) \ge \binom{n}{k}\binom{n-k+\ell}{n-k}$
- Proof: Consider only product by monomials supported on the variables that survived the derivative
- Claim: For any degree r polynomial f

$$\mu_k^\ell(f) \le \min\left\{\binom{n+k}{n}\binom{n+\ell}{n}, \binom{n+r-k+\ell}{n}\right\}$$

- Proof: First term bounds the possible number of different derivatives and different number of shifts. The second is the dimension of degree r-k+ℓ polynomials
- Fact: tight for a random f

Bounds for $\Sigma\Lambda\Sigma\Pi^{[b]}$ circuits

Claim: For deg(Q)=b: $\mu_k^{\ell}(Q^r) \leq \binom{n+(b-1)k+\ell}{n}$

Proof: order k' derivative of Q^r are of the form $Q^{r-k'} \cdot g$ where deg(g)=(b-1)k'. Hence, all polynomials in $\bar{x}^{\ell} \cdot \partial^k(Q^r)$ are $Q^{r-k} \cdot g$ where deg(g) $\leq$ (b-1)k+ℓ

Cor: f computed by $\Sigma\Lambda\Sigma\Pi^{[b]}$ with top fan-in s then

$$\mu_k^{\ell}(f) \leq s\binom{n+(b-1)k+\ell}{n}$$

Theorem [Kayal]: $\Sigma\Lambda\Sigma\Pi^{[b]}$ complexity of $x_1 \cdot \ldots \cdot x_n$ is $2^{\Omega(n/b)}$

Proof: Take ℓ= bn and k= $\varepsilon \cdot n/b$

Bounds for $\Sigma\Pi^{[a]}\Sigma\Pi^{[b]}$ circuits

Claim: For $\deg(Q_i)=b$: $\mu_k^{\ell}(Q_1 \cdot \cdots \cdot Q_a) \le \binom{a}{k}\binom{n+(b-1)k+\ell}{n}$

Proof: Each term is of the form $Q_{i1} \cdot \ldots Q_{i\{a-k'\}} \cdot g$ where
$\deg(g) = (b-1)k' + \ell$

Cor: f computed by $\Sigma\Pi^{[a]}\Sigma\Pi^{[b]}$ with top fan-in s then

$$\mu_k^{\ell}(f) \le s\binom{a}{k}\binom{n+(b-1)k+\ell}{n}$$

Cor: best bound is $\dfrac{\min\left\{\binom{n+k}{n}\binom{n+\ell}{n},\binom{n+r-k+\ell}{n}\right\}}{s\binom{a}{k}\binom{n+(b-1)k+\ell}{n}}$

Cor: For $a=b=\sqrt{r}$, $\ell = O\left(\frac{n\sqrt{r}}{\log n}\right)$, $k=\varepsilon\cdot\sqrt{r}$ a lower bound of $n^{\Omega(\sqrt{r})}$

Separating VP and VNP?

Just proved: Best possible lower bound is of $n^{\Omega(\sqrt{r})}$

Recall: homogeneous f in VP then f has a homogeneous $\Sigma\Pi^{[\sqrt{r}]}\Sigma\Pi^{[\sqrt{r}]}$ circuit of size $n^{O(\sqrt{r})}$

Dream approach for VP vs. VNP: Prove a lower bound of $n^{\Omega(\sqrt{r})}$ for a polynomial in VNP and improve the depth reduction just a little bit

Dream come true?

Theorem [Gupta-Kamath-Kayal-Saptharishi]:
$\mu_k^{\ell}(\mathrm{Perm}_n, \mathrm{Det}_n) \geq \binom{n+k}{2k}\binom{n^2-2k+\ell-1}{\ell}$,
bound tight for Det

Cor: their $\Sigma\Pi^{[\sqrt{n}]}\Sigma\Pi^{[\sqrt{n}]}$ complexity is $\exp(\Omega(\sqrt{n}))$

Goal: Better lower bounds for PERM (or f in VNP) and better depth reduction!

Theorem [Kayal-Saha-Saptharishi]: any $\Sigma\Pi^{[O(\sqrt{n})]}\Sigma\Pi^{[\sqrt{n}]}$ circuit for $NW_{\varepsilon\sqrt{n}}$ has size $n^{\Omega(\sqrt{n})}$

Great source of optimism, just improve depth reduction for VP

Well…

Theorem [Fourier-Limaye-Malod-Srinivasan]:
for $r \leq n^{\delta}$, IMM_r has $\Sigma\Pi^{[\sqrt{r}]}\Sigma\Pi^{[\sqrt{r}]}$ complexity $n^{\Omega(\sqrt{r})}$

Cor: Depth reduction cannot be improved

Theorem [Kumar-Saraf]:
$\forall \log n \ll t \leq r/40$ there is f computed by hom. $\Sigma\Pi\Sigma\Pi^{[t]}$ formula such that any hom. $\Sigma\Pi\Sigma\Pi^{[\frac{t}{20}]}$ circuit computing it requires size $n^{\Omega(\sqrt{r/t})}$

Cor: Depth reduction really cannot be improved

The NW polynomial

Exponent vectors form an error correcting code:

$$NW_k(x_{1,1}, \dots, x_{n,n}) = \sum_{\deg(p)<k} \prod_{i \in \mathbb{F}_n} x_{i,p(i)}$$

Main point [Chilara-Mukhopadhyay]: Monomials are "far away" hence, at most one monomial survives an order k derivative – easy to lower bound shifted partial dimension

Cor: For s=#Mon(NW_k) and N=n^2= #vars(NW_k)

number of distinct monomials in $\overline{\mathrm{x}}^{\ell} \cdot \partial^{=k}(NW_k)$ at least

$$s\binom{N+\ell}{N} - \binom{s}{2}\binom{N+\ell-(n-k)}{N}$$

Open: is {NW_k} complete for VNP?

Plan

- ✓ Survey of known lower bounds
- ✓ Some proofs:
  - ✓ General lower bounds
    - ✓ Strassen's nlog(n) lower bound
    - ✓ n^2 lower bound for ABPs/Formulas
  - ✓ Approximation method for $\Sigma\Pi\Sigma$ circuits over $\mathbb{F}_p$
  - ✓ Partial derivative method and applications
    - ✓ $\Sigma\Pi\Sigma$ circuits
    - ✓ Multilinear formulas
  - ✓ Shifted partial derivatives method
    - ✓ Application for $\Sigma\Pi\Sigma\Pi$ circuits

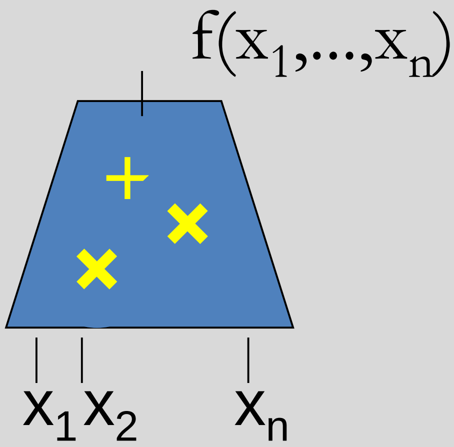
Polynomial Identity Testing (PIT)

Plan

- Basic definitions and motivation
- Universality of PIT
  - Equivalence to deterministic polynomial factorization
- Hardness vs. Randomness
  - PIT implies lower bounds and vice versa
- Survey of known results
- PIT for
  - $\Sigma\Pi$ circuits
  - $\Sigma\Lambda\Sigma$ circuits
  - $\Sigma\Pi\Sigma$ circuits – the rank method
- Summary

Polynomial Identity Testing

Input: Arithmetic circuit computing f
Problem: Is $f = 0$?

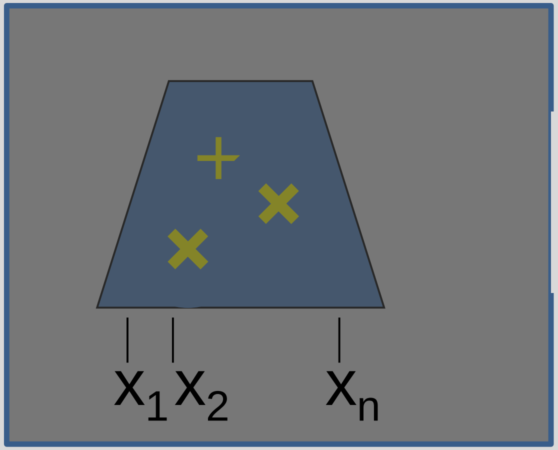


Note: $x^2 - x$ is the zero function over $\mathbb{F}_2$ but not the zero polynomial!

Black Box PIT = Hitting Set

Input: A Black-Box circuit computing f.

$(b_1,\ldots,b_n) \longrightarrow$



$\longrightarrow f(b_1,\ldots,b_n)$

Problem: Is $f = 0$?

[Schwart-Zippel-DeMilo-Lipton]: Evaluate at a random point

Goal: deterministic algorithm (a.k.a. Hitting Set):

Set H s.t. if $f \neq 0$ then $\exists a \in H$ s.t. $f(a) \neq 0$

Existence of a small hitting set

Infinite many circuits so counting arguments don't work

But, set of poly-size circuit generates a ``simple" variety (polynomial identified with vectors of coefficients)

Theorem [Heintz-Sieveking]: The set of n-variate degree-r polynomials computed in size s, defines a variety of dimension $(n+s)^2$ and degree $(sr)^\wedge(n+s)^2$

Theorem [Heintz-Schnorr]: A random subset of $[sr^2]$ of size $O((s+n)^2)$ is a hitting set whp.

Proof idea: Each "bad point" reduces dimension of variety by 1 (adds another constraint). Bound on degree is used when we reach dimension 0

Motivation

- Natural and fundamental problem
- Strong connection to circuit lower bounds
- Algorithmic importance:
  - Primality testing [Agrawal-Kayal-Saxena]
  - Randomized Parallel algorithms for finding perfect matching [Karp-Upfal-Wigderson, Mulmuley-Vazirani-Vazirani]
  - Deterministic algorithms for Perfect Matching in depth poly(log n) (and quasi-poly time) [Fenner-Gurjar-Thierauf, Svensson-Tarnawski]
- New approaches to derandomization in the Boolean setting
- PIT appears the most general derandomization problem

Motivation

- Natural and fundamental problem
- Strong connection to circuit lower bounds
- Algorithmic importance:
  - Primality testing [Agrawal-Kayal-Saxena]
  - Randomized Parallel algorithms for finding perfect matching [Karp-Upfal-Wigderson, Mulmuley-Vazirani-Vazirani]
  - Deterministic algorithms for Perfect Matching in depth poly(log n) (and quasi-poly time) [Fenner-Gurjar-Thierauf, Svensson-Tarnawski]
- New approaches to derandomization in the Boolean setting
- PIT appears the most general derandomization problem

Plan

- ✓ Basic definitions and motivation
- Universality of PIT
  - Equivalence to deterministic polynomial factorization
- Hardness vs. Randomness
  - PIT implies lower bounds and vice versa
- Survey of known results
- PIT for
  - $\Sigma\Pi$ circuits
  - $\Sigma\Lambda\Sigma$ circuits
  - $\Sigma\Pi\Sigma$ circuits – the rank method
- Summary

Universality of PIT

PIT is in coRP. Is it the most general language there?

Which other problems are in RP/BPP ???

Parallel algorithm for Perfect matching (PIT) in RNC

Languages coming from group theory

Example: Polynomial factorization

Given circuit for $f = f_1 \cdot f_2$ output circuits for f_1, f_2

A priori not clear such circuits exist

[Kaltofen]: Circuits exist and efficient randomized algorithm for constructing them!

[Kaltofen-Trager]: Also in the black-box model

Open: Are restricted models (bounded depth circuits, formulas, ABPs) close to taking factors?

Question: What is the cost of derandomizing polynomial factorization?

Factorization vs. PIT

Claim: $f(x)=0$ iff $f(x) + yz$ is reducible

Corollary: Deterministic factorization implies deterministic PIT

What about the other direction?

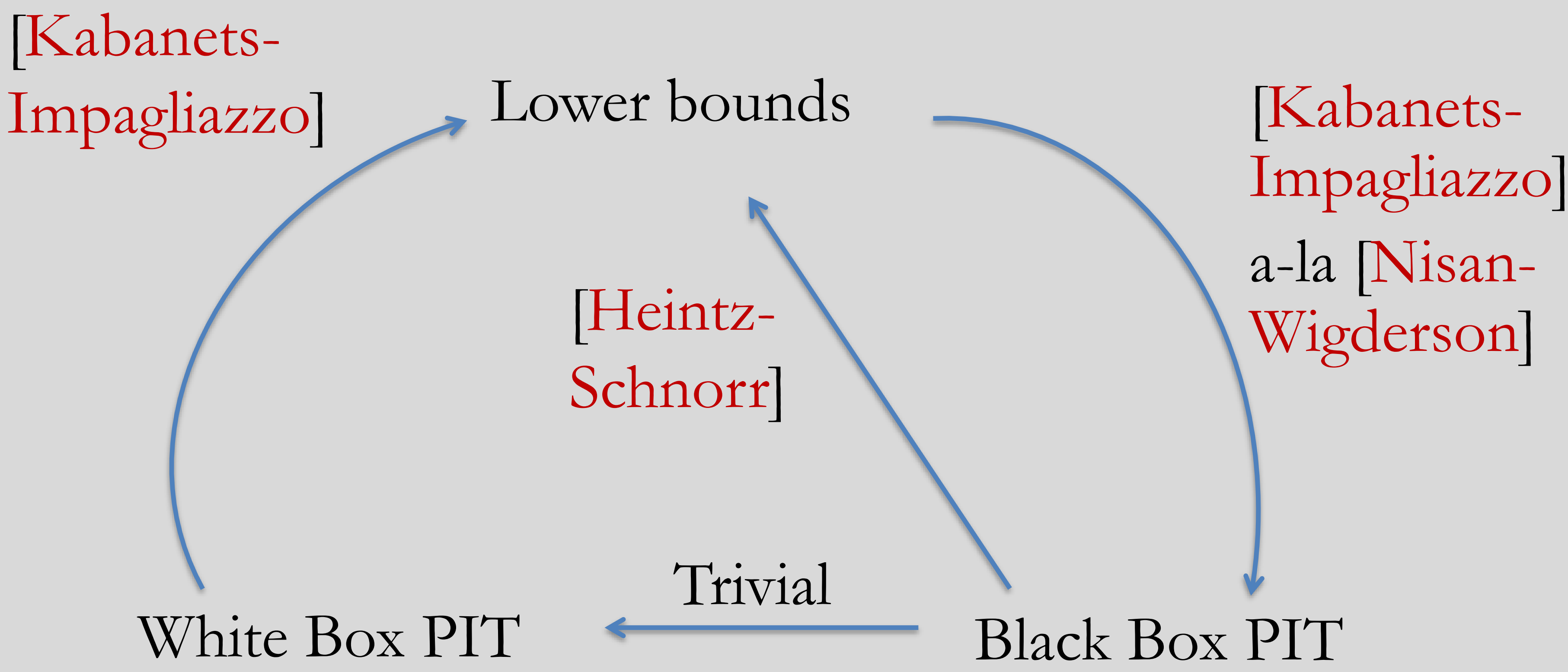
[S-Volkovich,Kopparty-Saraf-S]: Deterministic PIT implies deterministic factorization

Main idea: Carefully go over factorization algorithm and notice that randomization is used only to argue about nonzeroness of polynomials that have poly size circuits

Plan

- ✓ Basic definitions and motivation
- ✓ Universality of PIT
  - ✓ Equivalence to deterministic polynomial factorization
- Hardness vs. Randomness
  - PIT implies lower bounds and vice versa
- Survey of known results
- PIT for
  - $\sum\prod$ circuits
  - $\sum\wedge\sum$ circuits
  - $\sum\prod\sum$ circuits – the rank method
- Summary

Hardness vs. Randomness



Theorem: subexp PIT implies lower bounds, and

exp lower bounds ⇒ BB-PIT in quasi-P

BB PIT implies lower bounds

[Heintz-Schnorr]: BB PIT in P implies lower bounds

Proof: $|H|=n^{O(1)}$ hitting set for a class $\mathcal{C}$. Find a nonzero (multilinear) polynomial, f, with $\log|H|=O(\log n)$ variables vanishing on H. It follows that f requires exponential circuits from $\mathcal{C}$

Gives lower bounds for f computable in PSPACE

Conjecture [Agrawal]:

$H=\{(y_1,\ldots, y_n) : y_i=y^{k^i \bmod r}, y,k,r < s^{20}\}$ is a hitting set for size s circuits

WB PIT implies lower bounds

[Kabanets-Impagliazzo]: subexp WB PIT implies lower bounds

Proof idea:

- [Impagliazzo-Kabanets-Wigderson]: NEXP$\subseteq$P/poly $\Longrightarrow$ NEXP$\subseteq$P$^{\#P}$
- If PERM has poly-size circuits then guess one. Verify the circuit using PIT and self reducibility (expansion by row).
  Implies NEXP$\subseteq$ P$^{\#P}$ $\subseteq$ NSUBEXP in contradiction

[Kabanets-Impagliazzo]: lower bounds imply BB PIT

Proof idea: If f exponentially hard apply NW-design:

- $S_1,\ldots,S_n \subseteq [t=O(\log^2 n)]$
- $|S_i \cap S_j| \leq \log n$

Let $G(x)=(f(x|S_1),\ldots, f(x|S_n))$ map $\mathbb{F}^t$ to $\mathbb{F}^n$

Claim: If nonzero p has poly size circuit then $p \circ G$ nonzero

Proof: $p(y_1,\ldots,y_n)$ nonzero but $p(f(x|S_1),\ldots, f(x|S_n))$ zero.
Wlog $p(f(x|S_1),\ldots, f(x|S_{n-1}),y_n)$ nonzero.
Thus $(y_n-f(x|S_n))$ a factor of $p(f(x|S_1),\ldots, f(x|S_{n-1}),y_n)$.
By NW-design property polynomial has small circuit. By [Kaltofen], $(y_n-f(x|S_n))$ has small circuit in contradiction (pick t to match lower bound on f) ■

Evaluating G on $(r\cdot\deg(f))^t$ many points give a hitting set.

Extreme Hardness vs. Randomness

Theorem [Guo-Kumar-Saptharishi-Solomon]: Suppose for every s, ∃explicit hitting set of size $((s + 1)^k - 1)$ for k-variate polynomials of individual degree $\leq$ s that are computable by size s circuits

Then there is an explicit hitting set of size $s^{O(k^2)}$ for the class of s-variate polynomials, of degree s, that are computable by size s circuits

In other words: Saving one point over trivial hitting set for polynomials with O(1) many variables enough to solve PIT

Proof Idea: Hitting set $\Longrightarrow$ Hard polynomial $\Longrightarrow$ Hitting set (via a variant of the KI generator)

Plan

- ✓ Basic definitions and motivation
- ✓ Universality of PIT
  - ✓ Equivalence to deterministic polynomial factorization
- ✓ Hardness vs. Randomness
  - ✓ PIT implies lower bounds and vice versa
- Survey of known results
- PIT for
  - $\Sigma\Pi$ circuits
  - $\Sigma\Lambda\Sigma$ circuits
  - $\Sigma\Pi\Sigma$ circuits – the rank method
- Summary

Deterministic algorithms for PIT

$\sum\prod$ circuits (a.k.a., sparse polys), BB in poly time [BenOr-Tiwari, Grigoriev-Karpinski, Klivans-Spielman,…]

$\sum\Lambda\sum$ circuits, BB in $n^{\log\log(n)}$ time [Forbes-Saptharishi-S]

$\sum^{[k]}\prod\sum$ circuits

- BB in time $n^{O(k)}$ [Dvir-S,Kayal-Saxena,Karnin-S,Kayal-Saraf,Saxena-Seshadhri]
- Multilinear in sub-exponential time, for subexponential k [Oliveira-S-Volk] (implies nearly best lower bounds)

Multilinear $\sum^{[k]}\prod\sum\prod$ [Karnin-Mukhopadhyay-S-Volkovich, Saraf-Volkovich] BB in time $s^{poly(k)}$

Read-Once (skew) determinants [Fenner-Gurjar-Thierauf, Svensson-Tarnawski] BB in time $n^{(\log n)^2}$

Deterministic algorithms for PIT

Read-Once Algebraic Branching Programs

- White-Box in polynomial time [Raz-S]
- Black box in quasi-poly time [Forbes-S, Forbes-Saptharishi-S, Agrawal-Gurjar-Korwar-Saxena, Gurjar-Korwar-Saxena]
- Application to derandomization of Noether's normalization lemma, central in Geometric Complexity Theory program of Mulmuley

Read-k multilinear formulas / Algebraic Branching Programs [S-Volkovich, Anderson-van Melkebeek-Volkovich, Anderson-Forbes-Saptharishi-S-Volk]

- Subexponential WB for read-k ABPs
- Poly/quasi-poly for read-k Formulas (WB/BB)

Why study restricted models?

- [Agrawal-Vinay,Gupta-Kamath-Kayal-Saptharishi] PIT for $\sum\prod\sum$ (or homogeneous $\sum\prod\sum\prod$) circuits implies PIT for general depth
- roABPs: natural analog of Boolean roBP which capture RL
- Read-once determinants: new deterministic parallel algorithm for perfect matching.
- Gaining insight into more general questions:
  - Intuitively: lower bounds imply PIT
  - Multilinear formulas: super polynomial bounds [Raz] but no PIT algorithms
  - PIT gives more information than lower bounds.
- Interesting math: Extensions of Sylvester-Gallai type theorems

Plan

- ✓ Basic definitions and motivation
- ✓ Universality of PIT
  - ✓ Equivalence to deterministic polynomial factorization
- ✓ Hardness vs. Randomness
  - ✓ PIT implies lower bounds and vice versa
- ✓ Survey of known results
- PIT for
  - $\Sigma\Pi$ circuits
  - $\Sigma\Lambda\Sigma$ circuits
  - $\Sigma\Pi\Sigma$ circuits – the rank method
- Summary

PIT for ΣΠ circuits

$f = \Sigma_e c_e \Pi_i x_i^{e_i}$ with polynomialy many monomials

[Klivans-Speilman]: use $x_i \leftarrow y^{c_i}$ to map x-monomials 1-1

Set $c_i = c^i \bmod p$ (p prime larger than r)

$\bar{x}^{\bar{e}}$ is mapped to $y\hat{}\sum e_i c^i \pmod{p} = y\hat{}e(c) \pmod{p}$

If $\forall e \neq e'$, $e(c) \neq e'(c)$ then monomials are mapped 1-1

If s monomials then s^2 differences, each of degree $\leq r$, going over all choices of c in $[rs^2]$ gives a good map

Each possible c gives a low-degree univariate in y, evaluating at enough points gives the hitting set. Size $O(r^3 s^2)$.

PIT for $\Sigma\Lambda\Sigma$ circuits

Theorem: If leading monomial of f has m variables then dimension of partial derivatives of f is at least 2^m

Corollary: If f computed in size s then its leading monomial has at most log(ns) many variables.

Black Box PIT:

- "Guess" log(ns) variables. Set all other variables to zero.
- Interpolate resulting polynomial.

Theorem: Gives a hitting set of size $\deg^{\log(ns)}$.

Theorem [Forbes-Saptharishi-S]: By combining with PIT for roABP can get hitting set of size $s^{\log\log s}$.

Open: Polynomial time BB algorithm. ([Raz-S] gives WB)

PIT for ΣΠΣ circuits

How does an identity look like?

If $M_1 + \ldots + M_k = 0$ then

Multiplying by a common factor:

$$\Pi x_i \cdot M_1 + \ldots + \Pi x_i \cdot M_k = 0$$

Adding two identities:

$$(M_1 + \ldots + M_k) + (T_1 + \ldots + T_{k'}) = 0$$

How do the most basic identities look like?

Basic: cannot be "broken" to pieces (minimal) and no common linear factors (simple)

$\Sigma\Pi\Sigma$ identities

$C = M_1 + \ldots + M_k \qquad M_i = \Pi_{j=1\ldots d_i} L_{i,j}$

Rank: dimension of space spanned by $\{L_{i,j}\}$

Can we say anything meaningful about the rank?

Theorem [Dvir-S]: If $C \equiv 0$ is a basic identity then

$$\dim(C) \leq \mathrm{Rank}(k,r) = (\log(r))^k$$

White-Box Algorithm: find partition to sub-circuits of low dimension (after removal of g.c.d.) and brute force verify that they vanish.

Improved $(nr)^{O(k)}$ algorithm by [Kayal-Saxena]

Black-Box PIT for ΣΠΣ circuits

Black-Box Algorithm [Karnin-S]: Intuitively, if we project the inputs to a "low" dimensional space in a way that does not collapse the dimension below Rank(k,r) then identity should not become zero

Theorem [Gabizon-Raz]: ∃ "small" explicit set of D-dimensional subspaces $V_1,\ldots,V_m$ such that for every space of linear functions L, for most i:

$\dim(L|_{V_i}) = \min(\dim(L),D)$

In other words: the linear functions in L remain as independent as possible on V_i

Black-Box PIT for ΣΠΣ circuits

Corollary: $\forall i$ $C|_{V_i}$ has low "rank" $\Longrightarrow$ C has low "rank"

If C has high rank then by [Gabizon-Raz], for some i, $C|_{V_i}$ has high rank.

Black-Box PIT for $\Sigma\Pi\Sigma$ circuits

Corollary: $\forall i\ C|_{V_i}$ has low "rank" $\Longrightarrow$ C has low "rank"

Corollary: if $\forall$ i, $C|_{V_i} \equiv 0$ then C has structure (i.e. C is sum of circuits of low "rank")

If C is not a sum of low rank circuits then for some i, $C|_{V_i}$ is not a sum of low rank circuits. This contradicts the structural theorem.

Black-Box PIT for $\Sigma\Pi\Sigma$ circuits

Corollary: $\forall i\ C|_{V_i}$ has low "rank" $\Longrightarrow$ C has low "rank"

Corollary: if $\forall$ i, $C|_{V_i} \equiv 0$ then C has structure (i.e. C is sum of circuits of low "rank")

Theorem: if $\forall$i, $C|_{V_i} \equiv 0$ then $C \equiv 0$.

C is sum of low rank subcircuits $\Rightarrow$
$\exists V_i$ s.t. rank of subcircuits remain the same. $C|_{V_i}$ is zero $\Rightarrow$ each subcircuit vanishes on V_i $\Rightarrow$ subcircuits compute the zero polynomial.

Black-Box PIT for $\Sigma\Pi\Sigma$ circuits

Corollary: $\forall i\ C|_{V_i}$ has low "rank“ $\Longrightarrow$ C has low "rank"

Corollary: if $\forall$ i, $C|_{V_i} \equiv 0$ then C has structure (i.e. C is sum of circuits of low “rank”)

Theorem: if $\forall$i, $C|_{V_i} \equiv 0$ then $C \equiv 0$.

Algorithm: For every i, brute force compute $C|_{V_i}$

Time: $\mathrm{poly}(n)\cdot r^{\dim(V_i)} = \mathrm{poly}(n)\cdot r^{O(\mathrm{Rank}(k,r))}$

$\Sigma\Pi\Sigma$ identities

Lesson 1: depth 3 identities are very structured

Lesson 2: Rank is an important invariant to study

Improvements [Kayal-Saraf,Saxena-Seshadri]:

Finite field, $k \cdot \log(r) < \mathrm{Rank}(k,r) < k^3 \cdot \log(r)$

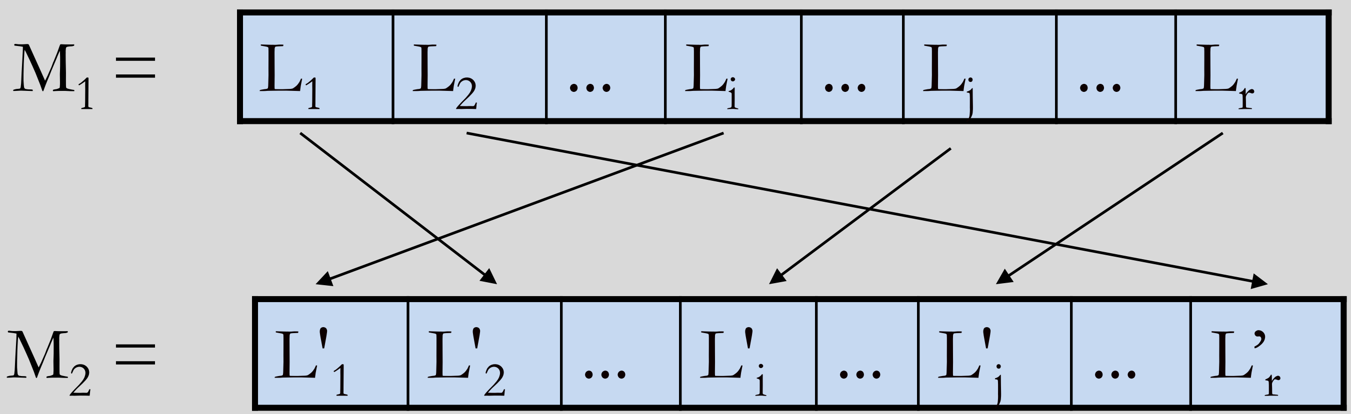
Over char 0, $k < \mathrm{Rank}(k,r) < k^2 \cdot \log(k)$

Improves [Dvir-S] + [Karnin-S] (plug and play)

Best PIT [Saxena-Seshadri]: BB-PIT in time $(nr)^{O(k)}$ (proof inspired by rank techniques)

Bounding the rank

Basic observation: Consider $C = M_1 + M_2$



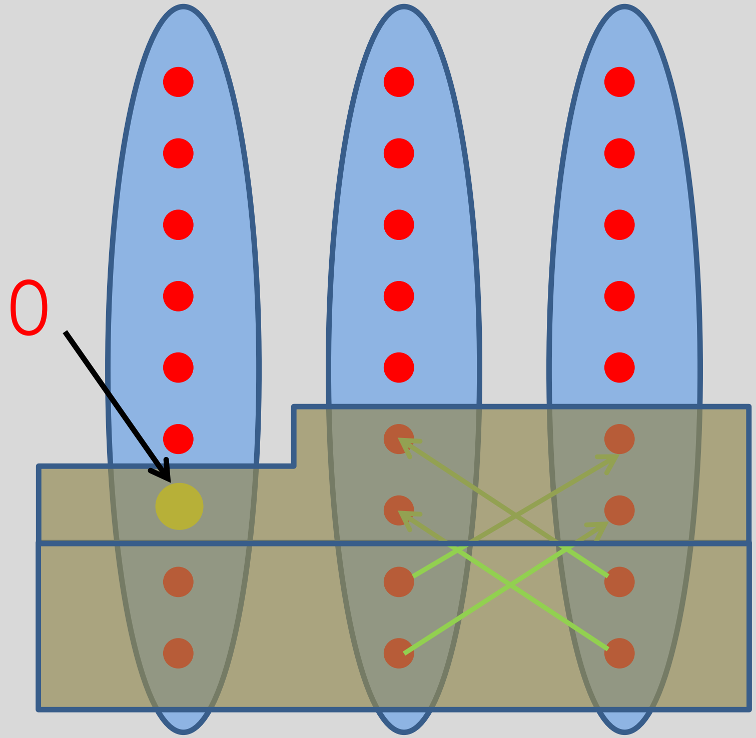
Fact: linear functions are irreducible polynomial.

Corollary: $C \equiv 0$ then M_1, M_2 have same factors.

Corollary: $\exists$ matching $i \to \pi(i)$ s.t. $L_i \sim L'_{\pi(i)}$

Bounding the rank

- Claim: $\mathrm{Rank}(3,r) = O(\log(r))$



Sketch: cover all linear functions in $\log(r)$ steps, where at m'th step:

- dim of cover is $O(m)$
- $\Omega(2^m)$ functions in span

Plan

- ✓ Basic definitions and motivation
- ✓ Universality of PIT
  - ✓ Equivalence to deterministic polynomial factorization
- ✓ Hardness vs. Randomness
  - ✓ PIT implies lower bounds and vice versa
- ✓ Survey of known results
- ✓ PIT for
  - ✓ $\sum\prod$ circuits
  - ✓ $\sum\wedge\sum$ circuits
  - ✓ $\sum\prod\sum$ circuits – the rank method
- Summary

Proofs – tailored for the model

Proofs usually use `weakness' inherent in model

- Depth 2: few monomials. Substituting y^{c_i} to x_i we can isolate different monomials
- Read-Once ABP: Polynomial has few linearly independent partial derivatives [Nisan]. Keep track of a basis for derivatives to do PIT
- $\Sigma\Pi\Sigma(k)$: setting a linear function to zero reduces top fan-in. If $k=2$ then multiplication gates must be the same. Calls for induction
- Multilinear $\Sigma\Pi\Sigma\Pi(k)$: in some sense `combination' of sparse polynomials and multilnear $\Sigma\Pi\Sigma(k)$
- Read-Once-Formulas: subformula of root contains ½ of variables

Summary

- PIT natural derandomization problem
- Equivalent to proving lower bounds
- Results for restricted models
- Open:
  - PIT for multilinear formulas
  - Improved PIT for multilinear depth 3
  - Poly time PIT for $\Sigma\Lambda\Sigma$ circuits
  - Closure of classes (ABPs, formulas) under factorization

Limitations and Approaches

Plan

- Limitations:
  - Limitations of (shifted) Partial Derivative Method
  - Natural Proofs for Arithmetic Circuits
  - The case of $\Sigma\Pi\Sigma$ circuits
- Approaches:
  - Matrix Rigidity
  - Elusive Polynomial Maps
  - Geometric Complexity Theory (GCT)
- Summary and open problems

Complexity Measure

Recall:

- $\mu_k(f) = \dim(\mathrm{span}(\partial^{=k}(f))$
- $\mu_k(f + g) \leq \mu_k(f) + \mu k(g)$
- $\mu_k(\ell^r) \leq 1$

Note: $\{\ell^r\}$ additive building blocks of $\Sigma\Lambda\Sigma$ circuits

Subadditivity implies: $\mathrm{size}_{\Sigma\Lambda\Sigma}(f) \geq \mu_k(f)/\mu_k(\ell^r)$

A barrier: when $\mu_k(f)$ cannot be much larger than μ_k(simple building block)

Abstracting the partial derivative method

(shifted) Partial derivative method: construct a huge matrix whose entries are linear functions in the coefficient of underlying polynomial. Rank of matrix is the measure

Example: f=xy+1

$$\begin{bmatrix} f \\ \partial f/\partial x \\ \partial f/\partial y \\ \partial^2 f/\partial x \partial y \end{bmatrix} = \begin{bmatrix} xy+1 \\ y \\ x \\ 1 \end{bmatrix} = \begin{array}{c} \begin{array}{cccc} xy & x & y & 1 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Abstract rank method

"Rank Method" = Linear map to matrices:

$$L : \text{Polynomials} \to \text{Mat}_{m\times m}(\mathbb{F})$$

Example: $\ell^r = (\sum a_i x_i)^r = \sum_{\bar{e}} \binom{r}{\bar{e}} \bar{a}^{\bar{e}} x^{\bar{e}}$

$$L(\ell^r) = \sum_{\bar{e}} \binom{r}{\bar{e}} \bar{a}^{\bar{e}} L(x^{\bar{e}}) = \sum_{\bar{e}} \binom{r}{\bar{e}} \bar{a}^{\bar{e}} M_{\bar{e}}$$

$L(\ell^r)$ = matrix with entries homogeneous polynomials in $\bar{a}$

Measure: $\mu_L(f) = \text{rank}(L(f))$

Lower bounds via abstract rank method

"Model" = Set of simple polynomials S that span all polynomials

Example: $S=\{\ell^r\}$ (for $\Sigma\Lambda\Sigma$ circuits)

Example: $S=\{\prod_{i=1}^{r} \ell_i\}$ (for $\Sigma\Pi\Sigma$ circuits)

Example : $S=\{g_{i1}\cdot g_{i2}\cdot g_{i3}\cdot g_{i4}\cdot g_{i5}\}$, $\deg(g_{ij}) \leq r/2$ (for general circuits)

Best lower bound in the model: $\text{size}_{\text{model}}(f) \geq \mu_L(f)/\mu_L(S)$

Barrier: when this ratio cannot be too large

Barrier on rank method

Theorem [Efremenko-Garg-Oliveira-Wigderson]: Rank method cannot prove more than $\Omega(n)^{\lfloor r/2 \rfloor}$ lower bound for homogeneous $\Sigma\Pi\Sigma$ circuits (similar bound also for $\Sigma\Lambda\Sigma$ circuits)

Cor: rank method cannot prove 8n lower bound on MM (best known lower bound is 3n-o(n) [S, Landsberg])

Note: for a random polynomial we expect $\Sigma\Pi\Sigma$ complexity to be $\Omega(n^{r-1}/r)$ (by counting degrees of freedom)

Recall: For the symmetric polynomial $\sigma_n^r(x)$ the lower bound obtained via partial derivative method is $\Omega(n^{r/2}/2^r)$

Proof Idea for $\Sigma\Lambda\Sigma$ circuits

Recall: $\mathrm{L}(\ell^{\mathrm{r}})$ is a matrix with entries homogeneous monomials in the coefficients of ℓ:

$$\mathrm{L}(\ell^{\mathrm{r}}) = \sum_{\bar{e}} \binom{r}{\bar{e}} \bar{a}^{\bar{e}} L(x^{\bar{e}}) = \sum_{\bar{e}} \binom{r}{\bar{e}} \bar{a}^{\bar{e}} M_{\bar{e}}$$

ϱ = maximum rank of $\mathrm{L}(\ell^{\mathrm{r}})$

= rank of $\sum_{\bar{e}} \binom{r}{\bar{e}} \bar{a}^{\bar{e}} M_{\bar{e}}$ as a matrix over $\mathbb{F}(\bar{a})$

(when entries viewed as polynomials in $\bar{a}$)

Maximal possible rank = maximal rank in span$\{\mathrm{L}(\ell^{\mathrm{r}})\}$

Main idea: show that $\mathrm{L}(\ell^{\mathrm{r}})$ are structured matrices and so is their span

Upper bounding the rank

Recall: $\mathrm{L}(\ell^{\mathrm{r}}) = \sum_{\bar{e}} \binom{r}{\bar{e}} \bar{a}^{\bar{e}} M_{\bar{e}}$ has rank at most ϱ

Can decompose over field of fractions (in $\bar{a}$)

$$L(\ell^r) = \sum_{i=1}^{\varrho} \frac{1}{p(\bar{a})} v_i(\bar{a}) \otimes u_i(\bar{a})$$

where $v_i(\bar{a}), u_i(\bar{a})$ vectors with entries polynomial in $\bar{a}$, and $p(\bar{a})$ is a polynomial

We now perform Strassen's trick to get rid of divisions!

$$L(\ell^r) = \sum_{i=1}^{\varrho} \frac{1}{p(\bar{a})} v(\bar{a}) \otimes u(\bar{a}) \qquad \text{w.l.o.g. } p(\bar{0}) = 1$$

$$L(\ell^r) = \sum_{i=1}^{\varrho} \frac{1}{1 - \tilde{p}(\bar{a})} v(\bar{a}) \otimes u(\bar{a})$$

$$= \sum_{i=1}^{\varrho} (1 + \tilde{p}(\bar{a}) + \tilde{p}^2(\bar{a}) + \tilde{p}^3(\bar{a}) + \cdots) v(\bar{a}) \otimes u(\bar{a})$$

Homogeneity implies

$$L(\ell^r) = H_r\left(\sum_{i=1}^{\varrho} \tilde{v}_i(\bar{a}) \otimes u(\bar{a}) \right)$$

$$L(\ell^r) = H_r\left(\sum_{i=1}^{\varrho} \tilde{v}_i(\bar{a}) \otimes u(\bar{a})\right)$$

$$= \sum_{i=1}^{\varrho}\sum_{j=0}^{r} H_j(\tilde{v}_i(\bar{a})) \otimes H_{r-j}(u_i(\bar{a}))$$

Main point: one of the vectors has degree at most $\left\lfloor \frac{r}{2} \right\rfloor$

Cor: summand is A+B where columns of A (rows of B) belong to a fixed space of dimension $\binom{n+\left\lfloor \frac{r}{2} \right\rfloor}{\left\lfloor \frac{r}{2} \right\rfloor}$

Plan

- Limitations:
  - ✓ Limitations of (shifted) Partial Derivative Method
  - Natural Proofs for Arithmetic Circuits
  - The case of $\Sigma\Pi\Sigma$ circuits
- Approaches:
  - Matrix Rigidity
  - Elusive Polynomial Maps
  - Geometric Complexity Theory (GCT)
- Summary and open problems

Natural proofs

[Razborov-Rudich] A property P of Boolean functions (truth tables) is natural if:

Useful against $\mathcal{C}$: If $P(f) = 1$ then we get a lower bound for circuits from $\mathcal{C}$ computing f

Constructivity: There is a $2^{poly(n)}$ sized circuit for computing $P(f)$ (input is truth table of f)

Largeness: For "many" functions f, $P(f) = 1$

[Razborov-Rudich]: All known lower bounds are natural

[Razborov-Rudich]: If PRFGs exist in $\mathcal{C}$ then no strong lower bounds for $\mathcal{C}$ (e.g. $\mathcal{C} = TC^0$)

Natural proofs barrier for arithmetic circuits?

Consider multilinear polynomials, given by list of coefficients

A property (polynomial) P is natural if

- Constuctivity: there is a $2^{\text{poly}(n)}$ sized arithmetic circuit for computing P(f)
- Usefulness: P(f) $\neq 0$ implies lower bounds on f

Note: All known proofs are natural

Example: having high partial derivative rank can be verified using determinant

Def: P is $\mathcal{D}$ natural against $\mathcal{C}$ if P computed by circuits from $\mathcal{D}$ and implies lower bounds for computing f in $\mathcal{C}$

Succinct hitting sets

Def: $\mathcal{C}$ is succinct hitting set for $\mathcal{D}$ if coefficient vectors of polynomials computed in $\mathcal{C}$ form a hitting set for $\mathcal{D}$

Note: We consider log(n)-variate polynomials in $\mathcal{C}$ and get hitting set for n-variate polynomials in $\mathcal{D}$

Observation [Grochow-Kumar-Saks-Saraf, Forbes-S-Volk]: No $\mathcal{D}$ natural property against $\mathcal{C}$, if $\mathcal{C}$ is succinct hitting set for $\mathcal{D}$

Conj: coefficient-lists of multilinear polynomial in VP hit VP (if true – no natural proofs for VP≠VNP)

Theorem [Forbes-S-Volk]: except of ro-Det all known hitting sets can be tweaked to multilinear-ΣΠΣ-succinct

Cor: Lower bounds on complexity of polynomials defining VP

Plan

- Limitations:
  - ✓ Limitations of (shifted) Partial Derivative Method
  - ✓ Natural Proofs for Arithmetic Circuits
  - – The case of $\Sigma\Pi\Sigma$ circuits
- Approaches:
  - – Matrix Rigidity
  - – Elusive Polynomial Maps
  - – Geometric Complexity Theory (GCT)
- Summary and open problems

Barrier for Lower Bounds for ΣΠΣ circuits

Recall: [S-Wigderson,Kayal-Saha-Tavenas] lower bound for ΣΠΣ circuits showed there exist $\Omega(n)$ many multiplication gates each of degree $\Omega(n)$ ($\Omega(n^2)$)

Proof idea: restrict to a subspace to make high degree gate vanish and then use (shifted) partial derivative measure on remaining circuit

Note: this approach cannot prove that there are more than n multiplication gates

Question: is there a reason for such a barrier?

Approximating polynomials

Def: g algebraically approximates f if $f(x)=g(\varepsilon,x) + \varepsilon\cdot h(\varepsilon,x)$, where monomials in h have degree > deg(f)

Theorem [Kumar]: every degree r polynomial can be approximated by $\Sigma\Pi\Sigma$ circuit with r+1 multiplication gates

"Cor": algebraic (continuous) measures cannot prove that more than r+1 multiplication gates are needed

Rationale: if a measure μ is small for every circuit with r+1 gates then it is small also for the limit. Thus, every polynomial has small μ complexity

Plan

- Limitations:
  - ✓ Limitations of (shifted) Partial Derivative Method
  - ✓ Natural Proofs for Arithmetic Circuits
  - ✓ The case of $\Sigma\Pi\Sigma$ circuits
- Approaches:
  - Matrix Rigidity
  - Elusive Polynomial Maps
  - Geometric Complexity Theory (GCT)
- Summary and open problems

Matrix Rigidity

Def: matrix A is (r,s)-rigid if we need to change more than s entries to reduce rank to r

Whenever $A=B+C$ either $\text{rank}(B) > r$ or C contains more than s nonzero entries

Theorem [Valiant]: If A is $(n/\log\log n, n^{1+\varepsilon})$-rigid then no linear circuit of size $O(n)$ and depth $O(\log n)$ can compute $f(x)=Ax$

Counting arguments: most matrices $(\Omega(n),O(n^2))$-rigid

Applications: Circuit complexity, lower bounds for data structures, locally decodable codes, …

Theorem [Friedman, Shokrollahi-Spielman-Stemann]: super regular matrices are $(r, n^2/r \cdot \log(n/r))$-rigid

Proof idea: Some rxr submatrix is not touched

Theorem [Alman-Williams, Dvir-Liu]: Hadmard like matrices not rigid enough

Theorem [Alman-Chen]: Using an NP oracle can construct $\left(2^{\log n^{1/4}}, \Omega(n^2)\right)$-rigid matrix

Note: new result by Orr et al.

Open: Find an explicit rigid matrix

Open: an explicit $(n-1,\Omega(n))$-matrix

Plan

- Limitations:
  - ✓ Limitations of (shifted) Partial Derivative Method
  - ✓ Natural Proofs for Arithmetic Circuits
  - ✓ The case of $\Sigma\Pi\Sigma$ circuits
- ✓ Approaches:
  - ✓ Matrix rigidity
  - – Elusive Polynomial Maps
  - – Geometric Complexity Theory (GCT)
- Summary and open problems

Elusive polynomial mappings

Def [Raz]: $f=(f_1,\ldots,f_m)$: $\mathbb{F}^n \to \mathbb{F}^m$ is (s,r)-elusive if for every

$g=(g_1,\ldots,g_m)$: $\mathbb{F}^s \to \mathbb{F}^m$, where $\deg(g_i) \leq r$,

$\mathrm{Image}(f) \not\subset \mathrm{Image}(g)$

Theorem [Raz]: If f is (s,2)-elusive for $m=n^{\omega(1)}$ and $s>m^{0.9}$, then super-polynomial lower bounds for f

Note: the moment curve (in 1 variable) is (m-1,1)-elusive for every m

Universal circuit

Def: circuit for degree r is in normal form if

– 2r alternating layers

– Edges go between layers

– Each constant gate has fan-out 1

Easy: each circuit can be made normal with poly blow up

Claim: for size s and degree r ∃ universal circuit U in x and $y=(y_1,\ldots,y_s)$ such that

– size(U) = poly(r,s)

– every size s normal circuit in x is obtained by assigning values to y vars

Circuits as polynomial maps

Note: Output of U is a polynomial in x,y. View it as a polynomial in x whose coefficients are polynomials in y

$\Rightarrow$ U defines a map $\Gamma: \mathbb{F}^s \rightarrow \mathbb{F}^m$ for $m = \binom{n+r}{n}$ mapping y to coefficient polynomials of x-monomials

Claim: Γ has degree 2r-1

Proof: each y variable used once in a layered circuit

Claim: if f has size s then f in image of Γ

Proof: follows from universality of U

Elusive maps

Cor: If $G: \mathbb{F}^n \to \mathbb{F}^m$ is (s,2r-1)-elusive then for some α, $G(\alpha)$ defines a hard polynomial (requires size > s)

Cor: if for every α, $G(\alpha)$ in VNP then can separate VP from VNP like that

Note: to claim about (s,2)-elusive maps need to use depth-reduction tricks

Plan

- Limitations:
  - ✓ Limitations of (shifted) Partial Derivative Method
  - ✓ Natural Proofs for Arithmetic Circuits
  - ✓ The case of $\Sigma\Pi\Sigma$ circuits
- ✓ Approaches:
  - ✓ Matrix Rigidity
  - ✓ Elusive Polynomial Maps
  - – Geometric Complexity Theory (GCT)
- Summary and open problems

Geometric complexity theory

Recall: want to show Perm is not a projection of Det

Action of matrices on polynomials: $(A\circ f)(x)=f(A\cdot x)$

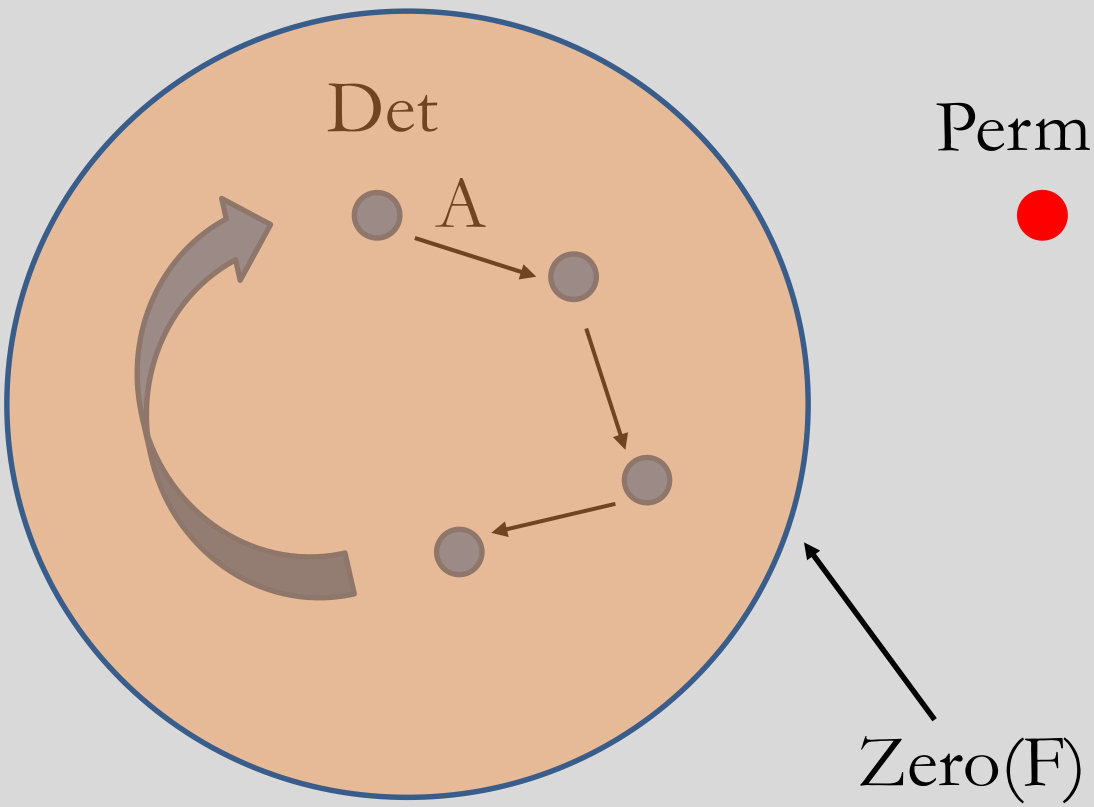
Goal: show Perm_n not in orbit of Det_m

Fact: the orbit of Det under matrices = closure of orbit of Det under GL (invertible matrices)

Fact: if Perm not in orbit then there is F (that takes as input coefficient vectors), such that F vanishes on (closure of) orbit of Det but not on Perm

Note: similar to Farkas lemma in linear programming

GCT approach [Mulmuley-Sohoni]: look for such polynomial using representation theory of GL


Det
A
Perm
Zero(F)

Why representation theory?

Separating F comes from a vector space $\mathcal{V}$ of polynomials acting on coefficient vectors

Can view GL action on coefficient vectors as action on polynomials from $\mathcal{V}$: $(A \circ F)(f) = F(A^t \circ f)$ (representation)

Consider all such F that vanish on the orbit of Det (Perm). They form a subrepresentation (linear subspace on which GL acts)

GCT approach: prove that these subrepresentations coming from the orbits of Det and Perm are different and conclude the existence of a separating F

Multiplicities

Conj [Mulmuley-Sohony]: Action of GL on orbit of Det has more irreducible representations than its action on orbit of Perm

Idea used by [Bürgisser-Ikenmeyer] to prove lower bounds for border rank of MM

Theorem [Ikenmeyer-Panova,Bürgisser-Ikenmeyer-Panova]: They have the same set of irreducible representation. Even $\Sigma\Lambda\Sigma$ circuits have the same set

New approach: prove that some irreducible representation appears more (higher multiplicity) over Perm than over Det

Recently implemented by [Ikenmeyer-Kandasamy] to separate a monomial from $\Sigma\Lambda\Sigma$

Summary

1. Basic definitions and structure results
2. Lower Bound techniques
3. PIT, hardness-randomness tradeoffs
4. Limitations, approaches

Model simpler than Boolean circuits, offers more chances to prove “big” results, classical math fits more naturally, many many open problems

Some more open problems

- Prove super polynomial lower bounds for bounded depth circuits over $\mathbb{F}_3$
- Prove super quadratic lower bounds for $\sigma_d(L_1,\ldots, L_m)$
- Exponential lower bound for multilinear formulas
- Separate multilinear and non-multilinear formula size
- Separate multilinear ABPs from multilinear circuits
- Super-poly lower bound for multilinear circuits
- Are formulas/ABPs/bounded-depth-circuits closed to taking factors?

Some more open problems

- What is the complexity of PIT: given H how hard is it to verify that H is a hitting set. Currently in EXPSPACE
- Results for read-once ABPs much better than in the Boolean world. Can techniques be used there?
- Theory of [Khovanskii] gives analogs of Bezout's theorem for sparse polynomials over $\mathbb{R}$ (sparsity replaces degree). Improve quantitative results. Would solve long standing open problems (PIT and algorithms)
- Reconstruction of arithmetic circuits
- …

Additional reading

[Bürgisser-Clausen-Shokrollahi]: Algebraic Complexity Theory

[S-Yehudayoff]: Arithmetic Circuits: a survey of recent results and open questions

[Saptharishi]: A selection of lower bounds in arithmetic circuit complexity

[Blaser-Ikenmeyer]: Introduction to geometric complexity theory (lecture notes)

Some more photos