Matrix Multiplication & Polynomial Identity Testing

Complexity Meetings

Robert Andrews

UIUC → ?

Matrix Multiplication

Problem: given $A, B \in \mathbb{C}^{n \times n}$, compute

$$C := A \cdot B$$

Question: how many arithmetic operations are necessary? sufficient?

Definition

$$\omega := \inf \{ \tau \in \mathbb{R} : A \cdot B \text{ can be computed in } O(n^\tau) \text{ operations} \}$$

Bounds on ω

$2 \leq \omega \leq 3$ By definition

≤ 2.81 [Strassen '69]

≤ 2.78 [Bini, Capovani, Romani '79]

≤ 2.522 [Schönhage '81]

≤ 2.3755 [Coppersmith, Winograd '90]

$\vdots$

≤ 2.37285 [Alman, Vassilevska W. '21]

Matrix Mult: Applications

- Computing the determinant
- Solving linear systems & linear programs
- QR decomposition, LUP decomposition
- Computing the characteristic polynomial
- Detecting k-cliques
- Recognizing context-free languages
- [Alman, Vassilevska W.] cited over 200 times in past 2 years...

Matrix Mult: Anti-Applications?

$\omega = 2 \Rightarrow$ fast algorithms for many problems

Question: Are there algorithmic consequences of $\omega > 2$?

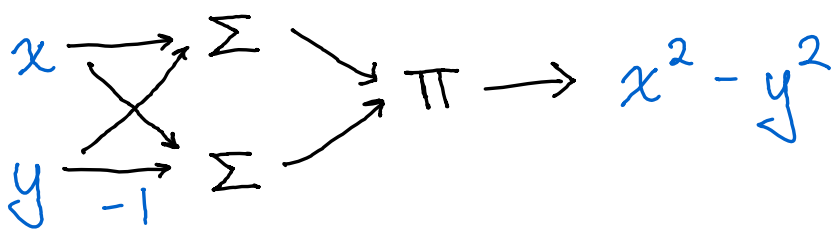
This talk: Yes!

(deterministic algorithms for polynomial identity testing)

Polynomial Identity Testing

Problem: given a polynomial $f(\bar{x}) \in \mathbb{C}[\bar{x}]$, decide if $f(\bar{x}) = 0$ as a polynomial
(assume $\deg(f) \leq n^{O(1)}$)

$f(\bar{x})$ given as an *algebraic circuit*:



- Easy to evaluate $f(\bar{x})$ at a point $\bar{a} \in \mathbb{C}^n$
- Too expensive to expand $f(\bar{x})$ as sum of monomials

PIT: Example 1

Is this identity true? $(i := \sqrt{-1})$

$$\prod_{k=1}^{4}\left(x^3 - i^k x + y + i^k \pi y^5 + z - i^k \sqrt{2}\, z\right)$$

$$+ \prod_{k=1}^{4}\left(x + i^k e x - \pi y^5 + i^k y - i^k z^2 + \sqrt{2}\, z\right)$$

$$+ \prod_{k=1}^{4}\left(-i^k x^3 - e x - i^k y - y + z^2 - i^k z\right)$$

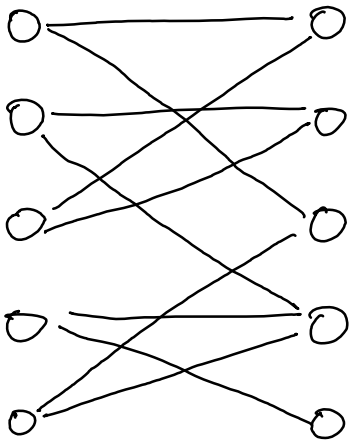
$$= 0?$$

PIT: Example 2

$$\det \begin{pmatrix} a & 0 & b & 0 & 0 \\ 0 & c & 0 & d & 0 \\ e & f & 0 & 0 & 0 \\ 0 & 0 & 0 & g & h \\ 0 & 0 & i & j & 0 \end{pmatrix} \stackrel{?}{=} 0$$

NO

Does this graph contain a perfect matching?



YES

Applications of PIT

- Parallel algorithms for perfect matching
- Primality testing
- Polynomial factorization, divisibility testing
- Proof of IP = PSPACE
- Testing equivalence of read-once branching programs
- Testing equality of compressed multidimensional strings

⋮

Solving PIT

PIT in coRP:

- Select any $S \subseteq \mathbb{C}$ of size $|S| \geq 2\deg(f)$
- Sample a random point $\bar{\alpha} \in S^n$
- Report $f(\bar{x}) = 0$ iff $f(\bar{\alpha}) = 0$

If $f(\bar{x}) = 0$, $\text{Prob}[\text{error}] = 0$

If $f(\bar{x}) \neq 0$, use Schwartz-Zippel lemma:

$$\text{Prob}[\text{error}] \leq \frac{\deg(f)}{|S|} \leq 1/2$$

DERANDOMIZING PIT

Question: can we design efficient deterministic algorithms for PIT?

[Kabanets-Impagliazzo '04, ...]: lower bounds for perm(X) $\Rightarrow$ derandomize PIT

[Heintz-Schnorr '80, ...]: derandomize PIT $\Rightarrow$ new circuit lower bounds

PIT algorithms are known for many circuit classes where we have lower bounds

(read-once branching programs, low-depth circuits, ...)

Main Result (Win-Win)

Theorem [A.]

At least one of the following is true.

(1) $\omega = 2$

(2) $\exists \varepsilon, \delta > 0$ such that PIT for circuits with $O(n^{1+\delta})$ multiplication gates can be solved deterministically in $\exp(O(n^{1-\varepsilon}))$ time.

(Naive algorithm for (2) runs in $\exp(O(n \log n))$ time.)

Main Result (Hard → Rand)

Theorem [A.]

Let $C(\bar{x})$ be a circuit with s multiplication gates. Then one can test if $C(\bar{x}) = 0$ deterministically in time

$$\exp\left(O\left(n^{1/2} s^{1/\omega} \log n\right)\right) \cdot s$$

$\omega = 2.03$
$s = n^{1.01}$
$\Rightarrow$
Test $C(\bar{x}) = 0$ in $\exp\left(O(n^{0.998})\right)$ time

Main Result (Hard → Rand)

Theorem [A.]

Let $C(\bar{x})$ be a circuit with s multiplication gates. Then one can test if $C(\bar{x}) = 0$ deterministically in time

$$\exp\left(O\left(n^{1/2}\,\underline{R}^{-1}(s)\,\log n\right)\right)\cdot s$$

$\underline{R}(n)$:= border rank of $n \times n$ matrix mult.

Theorem [Bini '80]: $\omega = \lim_{n\to\infty} \log_n\left(\underline{R}(n)\right)$

Possible that $\omega = 2$ but $\underline{R}(n) = n^2 \log^{100} n$

Unconditional PIT

Fact: $R(n) \geq n^2$

Corollary [A.]

Circuits with s multiplication gates can be tested in $\exp(O(\sqrt{ns} \cdot \log n)) \cdot s$ time.

Circuits with few multiplications are not obviously weak: they can compute dense polynomials using $O(\log n)$ multiplications via repeated squaring

$$(1 + x_1 + \cdots + x_n)^n$$

Better Parameters?

This algorithm is non-trivial for $s < \frac{1}{192} n$.

Can we do better? Not by much!

[Baur & Strassen '83]: any circuit computing

$$x_1^n + \cdots + x_n^n$$

must use $\Omega(n \log n)$ multiplications

Non-trivial black-box PIT for $s \gg n \log n \Rightarrow$ new circuit lower bounds!

Hitting Set Generators

Let $\mathcal{C} \subseteq \mathbb{F}[\bar{x}]$ be a set of polynomials.
A polynomial map

$$G : \mathbb{F}^{\ell} \to \mathbb{F}^{n}$$

$$(y_1, \ldots, y_\ell) \mapsto (G_1(\bar{y}), \ldots, G_n(\bar{y}))$$

is a hitting set generator for $\mathcal{C}$ if

$$\forall f(\bar{x}) \in \mathcal{C}, \quad f(\bar{x}) \neq 0 \Rightarrow f(G(\bar{y})) \neq 0 .$$

Parameters:

- seed length ℓ
- degree $\deg(G) := \max_i \deg(G_i(\bar{y}))$

Hitting Set Generators

A generator leads to faster PIT for $\mathcal{C}$:

Algorithm 1

Test $f(\bar{x}) = 0$ by brute force

evaluations

$(\deg(f) + 1)^n$

Algorithm 2

Test $f(g(\bar{y})) = 0$ by brute force

$(\deg(f) \cdot \deg(g) + 1)^\ell$

Proof Attempt #1

$n \times n$ matrices of variables

Assume $\omega = 2.1$. [Baur-Strassen '83] $\Rightarrow$ $\mathrm{trace}(XYZ)$ requires circuits of size $\Omega(n^{2.1})$

[Kabanets-Impagliazzo '04]: build a generator G_{KI} s.t.

if $f(\bar{x})$ can be computed in size $O(n)$

and $f \circ G_{KI} = 0$,

then $\mathrm{trace}(XYZ)$ has $\tilde{O}(n^6)$-size circuits

Issue: the lower bound is too weak!

Proof Attempt #2

Idea: white-box hardness versus randomness!

$\omega = 2.1 \Rightarrow$ linear algebra is hard

Goal: design a generator G such that breaking G requires linear algebra

Proof Attempt #2

Idea: white-box hardness versus randomness!

$\omega = 2.1$ [Baur-Strassen '83] $\Rightarrow$ $\det_n(X)$ requires $\Omega(n^{2.1})$ multiplications

$\Rightarrow$ $O(N)$-size circuits cannot compute the $\sqrt{N} \times \sqrt{N}$ determinant.

Since $\det_n(A) = 0$ iff $\text{rank}(A) < n$, singular matrices should look pseudorandom to $O(N)$-size circuits

Detecting Singular Matrices

Let $f(X) \in \mathbb{C}[X]$.

Fact: if $f(A) = 0$ for all singular $A \in \mathbb{C}^{n \times n}$,
then $f(X) = \det(X) \cdot g(X)$ for some $g(X) \in \mathbb{C}[X]$

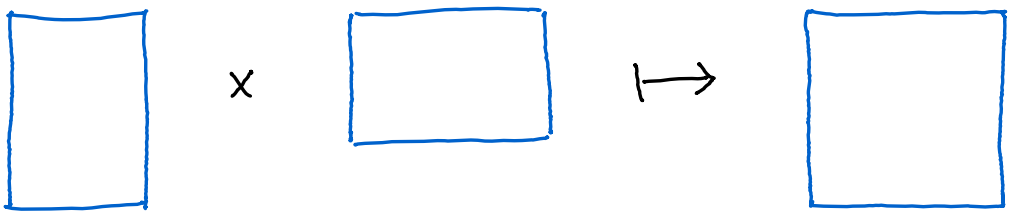
Q1) If $\omega = 2.1$, does $\det(X) \cdot g(X)$ require $\Omega(n^{2.1})$ multiplications?

Q2) How to evaluate at all singular matrices?

The Generator

Define

$$G(Y,Z): \mathbb{C}^{\sqrt{N}\times(\sqrt{N}-1)} \times \mathbb{C}^{(\sqrt{N}-1)\times\sqrt{N}} \to \mathbb{C}^{\sqrt{N}\times\sqrt{N}}$$

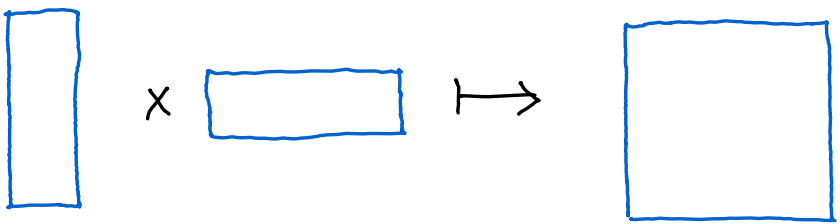


Fact: $\text{image}(G) = \{\text{singular matrices}\}$

Lemma: $f \circ G = 0 \iff f(X) = \det(X) \cdot g(X)$

The Generator

$G_r(Y,Z) : \mathbb{C}^{\sqrt{N} \times r} \times \mathbb{C}^{r \times \sqrt{N}} \to \mathbb{C}^{\sqrt{N} \times \sqrt{N}}$



Fact: $\text{image}(G) = \{\text{rank} \leq r \quad \text{matrices}\}$

Lemma: $f \circ G_r = 0$ iff

$(r+1)\times(r+1)$ submatrix

$f(X) = g_1(X) \cdot \det\left(\quad\right) + g_2(X) \cdot \det\left(\quad\right) + \cdots$

ROADMAP

$\omega = 2.1 \Rightarrow^{*}$ need $\Omega(r^{2.1})$ multiplications to compute $f(x)$ s.t. $f \circ G_r = 0$

$\Rightarrow$ G_r with $r = N^{0.49}$ is a generator that hits size $O(N)$ circuits

$\Rightarrow$ G_r has seed length $N^{0.99}$

$\Rightarrow$ can test size $O(N)$ circuits in $\exp(O(N^{0.99} \log N))$ time

ROADMAP

$\omega = 2.1 \Rightarrow^{*}$ need $\Omega(r^{2.1})$ multiplications to compute $f(x)$ s.t. $f \circ G_r = 0$

$\Rightarrow$ G_r with $r = N^{0.49}$ is a generator that hits size $O(N)$ circuits

$\Rightarrow$ G_r has seed length $N^{0.99}$

$\Rightarrow$ can test size $O(N)$ circuits in $\exp(O(N^{0.99} \log N))$ time

Lifting the Lower Bound

[Baur-Strassen '83]

$\omega = 2.1$ $\Longrightarrow$ $\text{trace}(YZW)$ requires $\Omega(r^{2.1})$ multiplications

($r \times r$ matrices)

This work $\Longrightarrow$ all polynomials of the form

$$\prod_{i=1}^{r} \det\left(X_{\le i, \le i}\right)^{a_i} \quad \text{with } a_r \ge 1$$

require $\Omega(r^{2.1})$ multiplications

[Andrews-Forbes '22] $\Longrightarrow$ all $f(X)$ vanishing on $G_r(Y, Z)$ requires $\Omega(r^{2.1})$ multiplications

Reducing $\mathrm{tr}(YZW)$ to $\det(X)$

$$M := \begin{pmatrix} I_n & Y & & \\ & I_n & Z & \\ & & I_n & W \\ \varepsilon \cdot I_n & & & I_n \end{pmatrix}$$

$$\det(M) = 1 + \varepsilon \cdot \mathrm{tr}(YZW) + O(\varepsilon^2)$$

$$\Rightarrow \mathrm{tr}(YZW) = \lim_{\varepsilon \to 0} \frac{\det(M) - 1}{\varepsilon}$$

$$= \lim_{\varepsilon \to 0} \frac{\det(M)^2 - 1}{2\varepsilon}$$

$$\vdots$$

Open Questions

(1) Does this result hold over fields of positive characteristic?

(2) Can the PIT algorithm be bootstrapped? Can the lower bound be magnified?

(3) Is it possible to design a PIT algorithm that matches the Baur–Strassen bound?

Thank you!