Introduction to algebraic complexity theory and how geometry enters

Christian Ikenmeyer
Agenda

1. Algebraic complexity theory
2. Geometry
Agenda

1 Algebraic complexity theory

2 Geometry
Algebraic algorithms

- Fast Fourier transform, fast matrix multiplication, ...

- Solving systems of linear equations

- Solving systems of polynomial equations: Gröbner bases

- Coding Theory: Reed-Muller codes, ...

- Number theory: Euclidean algorithm, Chinese Remainder Theorem, ...

Analyzing running time of algebraic algorithms:

- Number of arithmetic operations

- Size/growth/precision of the numbers
Arithmetization

Computation modulo 2: The field $F_2$

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Boolean "and" Boolean "xor"

Translate Boolean circuit using \{and, or, not\} $\leftrightarrow$ algebraic circuit using \{+, *\}:

During the translation the circuit only grows in size by at most a factor of 4.
Example

\[(x_1 \text{ and } x_3) \text{ or } (x_2 \text{ and } \text{not}(x_3))\]

\[x_1 x_2 x_3^2 + x_1 x_2 x_3 + x_1 x_3 + x_2 x_3 + x_2\]
Infinite fields

- Algebraic circuits naturally compute a polynomial
- Problem: Different polynomials can give the same function:
  \[ x^2y + x = xy + x^2 \quad \text{for all } x, y \in \mathbb{F}_2, \]
  but \( \text{coeff}_{xy}(x^2y + x) = 0 \neq 1 = \text{coeff}_{xy}(xy + x^2) \).
- The situation is better over infinite fields (for example \( \mathbb{C} \)):

**Lemma**

Over an infinite field, two polynomials compute the same function iff they have the same coefficient list.

Proof: Simple induction and polynomial division.
“Algebraic P vs NP”

The determinant polynomial:

\[ \det_m = \sum_{\pi \in S_m} \operatorname{sgn}(\pi) \prod_{i=1}^{m} x_{i,\pi(i)} \]

The permanent polynomial:

\[ \operatorname{per}_m = \sum_{\pi \in S_m} \prod_{i=1}^{m} x_{i,\pi(i)} \]

Assume from now on \( \operatorname{char} \mathbb{F} \neq 2 \), because otherwise \( \det_m = \operatorname{per}_m \).

Def.: The algebraic circuit size \( a(\operatorname{per}_m) \) is the smallest size of an algebraic circuit computing \( \operatorname{per}_m \).

Algebraic P vs NP conjecture \( (\text{VP} \neq \text{VNP}, \text{Valiant 1979}) \)

\( a(\operatorname{per}_m) \) is not polynomially bounded.
Determinants instead of circuits

Theorem (Valiant 1979)

Every multivariate polynomial $f$ can be written as the determinant of a matrix whose entries are polynomials of degree $\leq 1$.

Example: $f := y + 2x + xz + 2xy - x^2z = \det \begin{pmatrix} x & y & 0 \\ -1 & z + y + 2 & x \\ 1 & z & 1 \end{pmatrix}$

Def.: Required size of the matrix is called the **determinantal complexity** $dc(f)$.

In the example we have $dc(f) \leq 3$.

Valiant’s determinant vs permanent conjecture

$dc(\text{per}_m)$ is not polynomially bounded.

This is implied by $\text{VP} \neq \text{VNP}$.
Resources in algebraic computation

\[ \det \begin{pmatrix} x + 1 & y \\ -1 & x + 1 \end{pmatrix} \]

- Computes \( \sum_{s-t-path \ p} \prod_{edge \ e \in p} \text{label}(e) \)
- \( w(p) := \) the smallest width of an ABP computing \( p \).

**Theorem (Toda 1991)**

\( \text{dc}(p) \) and \( w(p) \) are polynomially related.
Definition p-family

A p-family is a sequence $(f_n)_{n \in \mathbb{N}}$ of polynomials such that:
- The number of variables is polynomially bounded
- The degree is polynomially bounded

- $\text{VF} := \{ \text{p-family whose formula size is polynomially bounded} \}$
- $\text{VBP} := \{ \text{p-family whose dc (or w) is polynomially bounded} \}$
- $\text{VP} := \{ \text{p-family whose circuit size is polynomially bounded} \}$

$(f_n) \in C$ is complete for $C$ if $\forall (g_m) \in C$ there exists a polynomially bounded $s$ and linear polynomials $\ell_i$ such that

$$\forall m : g_m = f_s(m)(\ell_1, \ell_2, \ldots).$$

For example, $(\det_n)$ is VBP-complete.

Example: $(x_1 x_2 \cdots x_n) \in \text{VBP}$, because $\det(\text{diag}(x_1, x_2, \ldots, x_n)) = x_1 x_2 \cdots x_n$.

$\text{IMM}_r^{(d)} := (x_1,1,1 \ x_1,2,1 \cdots x_1,r,1) \begin{pmatrix} x_1,1,2 & \cdots & x_1,r,2 \\ \vdots & \ddots & \vdots \\ x_r,1,2 & \cdots & x_r,r,2 \end{pmatrix} \cdots \begin{pmatrix} x_1,1,d-1 & \cdots & x_1,r,d-1 \\ \vdots & \ddots & \vdots \\ x_r,1,d-1 & \cdots & x_r,r,d-1 \end{pmatrix} \begin{pmatrix} x_1,1,d \\ \vdots \\ x_1,r,d \end{pmatrix}$

- $\text{IMM}^{(n)}_3$ isVF-complete [Ben-Or, Cleve 1988].
- $\text{IMM}^{(n)}_n$ is VBP-complete.
- There is no equally nice VP-complete p-family known.
Definition **VNP**

A p-family \((f_n)\) is in **VNP** if there exists a p-family \((g_n) \in \text{VP}\) and polynomially bounded functions \(r, s, t\) such that

\[
\forall n : f_n = \sum_{b \in \{0,1\}^r(n)} g_{t(n)}(x_1, \ldots, x_{s(n)}, b_1, \ldots, b_{r(n)})
\]

For example,

\[
\text{per}_n = \sum_{b \in \{0,1\}^{n^2}} C(b) \prod_{1 \leq i, j \leq n} (b_{i,j}(x_{i,j} - 1) + 1),
\]

where \(C\) is the arithmetization of a Boolean circuit checking if \(b\) is a permutation matrix.

- One can also take \((g_n) \in \text{VF}\) and it gives the same class: **VNP = VNF**.
- One can also take \((g_n)\) to be just a polynomially long product of linear polynomials (Bringmann-I-Zuiddam 2018)

Valiant 1979:

- The permanent p-family \((\text{per}_n)\) is **VNP**-complete.
Efficient computation:
- $\text{VF} \subseteq \text{VBP} \subseteq \text{VP}$
- “$\text{VBP} = \text{linear algebra}”$ (determinant, iterated matrix multiplication)

Efficiently definable (“explicit polynomials”):
- $\text{VNP}$
- “$\text{VNP} = \text{combinatorics/counting}”$ (Cycle covers, permanent)

**Valiant’s conjectures**

- $\text{VF} \neq \text{VNP}$
- $\text{VBP} \neq \text{VNP}$, determinant vs permanent, linear algebra vs counting
- $\text{VP} \neq \text{VNP}$
Valiant’s conjectures

\[
\text{VF} \neq \text{VNP} \\
\text{VBP} \neq \text{VNP}, \text{ determinant vs permanent, linear algebra vs counting} \\
\text{VP} \neq \text{VNP}
\]

These algebraic conjectures are “easier” than the Boolean ones:

\[
\text{PH} \neq \Sigma_2^P \quad \text{Karp-Lipton 1982} \quad \Rightarrow \quad \text{NP} \not\subset \text{P/poly} \quad \text{B"urgisser 2000} \quad \Rightarrow \quad \text{VP} \neq \text{VNP} \quad \Rightarrow \quad \text{VBP} \neq \text{VNP} \quad \Rightarrow \quad \text{VF} \neq \text{VNP}
\]

B"urgisser’s result works

- over finite fields, and
- over \( \mathbb{C} \) (if the generalized Riemann hypothesis is true).
Agenda

1 Algebraic complexity theory

2 Geometry
We work with homogenized algebraic branching programs!
In fact, inhomogeneous set-ups can lead to weird behavior in the representation theory (Landsberg-Kadish 2012, I-Panova 2015, Bürgisser-I-Panova 2015)

For a fixed degree $d$ and number of variables $n$ and a complexity bound $r$, study the set

$$X_r := \{ f \in \mathbb{C}[x_1, \ldots, x_n]_d \mid w(f) \leq r \}.$$ 

$$X_1 \subseteq X_2 \subseteq X_3 \subseteq \cdots \subseteq X_{\text{max}} = \mathbb{C}[x_1, \ldots, x_n]_d$$

$$\text{IMM}^{(d)}_r := (x_{1,1,1} \ x_{1,2,1} \cdots x_{1,r,1}) \begin{pmatrix} x_{1,1,2} & \cdots & x_{1,r,2} \\ \vdots & \ddots & \vdots \\ x_{r,1,2} & \cdots & x_{r,r,2} \end{pmatrix} \cdots \begin{pmatrix} x_{1,1,d-1} & \cdots & x_{1,r,d-1} \\ \vdots & \ddots & \vdots \\ x_{r,1,d-1} & \cdots & x_{r,r,d-1} \end{pmatrix} \begin{pmatrix} x_{1,1,d} \\ \vdots \\ x_{1,r,d} \end{pmatrix}$$

**VBP completeness with homogenization gives:**

- If $f \in X_r$, then $f(A \overrightarrow{x}) \in X_r$ for any linear map $A$.
- Every $f \in X_r$ can be obtained via a linear map $A$ as $f = \text{IMM}^{(d)}_r(A \overrightarrow{x})$

For example, if $f = x_1^3 + x_1x_2x_3$, and $A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$, then $f(A \overrightarrow{x}) = f(x_2, x_1, x_1 + x_2) = x_2^3 + x_1^2x_2 + x_1x_2^2$. 
$X_r := \{ f \in \mathbb{C}[x_1, \ldots, x_n]_d \mid w(f) \leq r \}$.

- Goal: find more useful mathematical structure on $X_r$.
- $X_r$ is closed under base changes: Changing input variables to linear combination comes at no extra cost.
- But $X_r$ is lacking a crucial property: It is not topologically closed.

Example from now on via Waring rank $\text{WR}$ instead of $w$.

For a homogeneous degree $d$ polynomial $f$ the **Waring rank** $\text{WR}(f)$ is defined as the smallest $r$ such that there exist homogeneous linear $\ell_i$ such that $f = \sum_{i=1}^r (\ell_i)^d$.

Not entirely obvious at first: $\text{WR}(f)$ is always finite.
For example:
$12x^3y = (x+y)^4 + i^3(x+iy)^4 + i^2(x+i^2y)^4 + i(x+i^3y)^4$, hence $\text{WR}(x^3y) \leq 4$. In fact, $\text{WR}(x^{d-1}y) = d$. 
\[ 12x^3y = (x + y)^4 + i^3(x + iy)^4 + i^2(x + i^2y)^4 + i(x + i^3y)^4. \]

The **border Waring rank** \( \text{WR}(f) \) is defined as the smallest \( r \) such that \( f \) can be approximated arbitrarily closely by polynomials of Waring rank \( \leq r \). For example, \( \text{WR}(x^3y) \leq 2 \).

For a space of polynomials \( V = \mathbb{C}[x_1, \ldots, x_n]_d \), the elements of \( \mathbb{C}[V] \) are called **meta-polynomials**.

**Example:** For \( ax^2 + bxy + cy^2 \), the discriminant \( b^2 - 4ac \) is a meta-polynomial.

**Theorem (algebraic geometry)**

The set \( \overline{W}_r = \{ f \mid \text{WR}(f) \leq k \} \) is an algebraic variety, i.e., there exist finitely many meta-polynomials \( \Delta_1, \ldots, \Delta_N \) with

\[ f \in \overline{W}_r \iff \Delta_1(f) = \Delta_2(f) = \cdots = \Delta_N(f) = 0 \]

We conclude that we know how complexity lower bounds must look like:

**Theorem:** If \( f \notin \overline{W}_r \), then there exists a meta-polynomial \( \Delta \) with \( \bullet \Delta(\overline{W}_r) = \{0\} \) and \( \bullet \Delta(f) \neq 0 \).
\[ A := \mathbb{C}[x, y]_2 = \langle x^2, xy, y^2 \rangle. \]

Every element in \( A \) can be represented as \( ax^2 + bxy + cy^2 \).

- \( X := \{ f \in A \mid \exists \alpha, \beta \in \mathbb{C} : f = (\alpha x + \beta y)^2 \} \) set of Waring rank 1 polynomials
- \( f \in X \) iff \( \Delta(f) = b^2 - 4ac = 0 \).
- To prove \( \text{WR}(f) \geq 2 \) we compute \( \Delta(f) \neq 0 \)
Topological closures of algebraic complexity classes

\[ \text{WR} \rightarrow \text{WR} \]

Analogously, we can allow such approximations

- for formulas,
- for algebraic branching programs,
- for circuits,
- or for hypercube summations of circuits.

The corresponding complexity classes are

- \( \text{VF} \subseteq \overline{\text{VF}} \),
- \( \text{VBP} \subseteq \overline{\text{VBP}} \),
- \( \text{VP} \subseteq \overline{\text{VP}} \),
- \( \text{VNP} \subseteq \overline{\text{VNP}} \).

Wide open question: Is \( \overline{\text{VF}} \subseteq \text{VNP} \)?
Let $X_r := \{ f \in \mathbb{C}[x_1, \ldots, x_n]_d \mid w(f) \leq r \}.$

- To find meta-polynomials $\Delta$ for $\overline{X_r}$, instead of studying $\overline{X_r}$ directly, one can study its **coordinate ring**, i.e., ring of polynomial functions restricted to $\overline{X_r}$.

- Representation theory helps to study the coordinate ring using the weights of the $\text{GL}_n$: a finer variant of a degree for meta-polynomials.


**Conclusion**

- We do not know how powerful approximations in algebraic complexity theory are, but
- if we allow approximations, then all complexity lower bounds come from meta-polynomials, and this opens a wide array of tools from algebraic geometry and representation theory.

Thank you very much for your attention!