On the Complexity of Symmetric Polynomials

Markus Bläser[1] [2]Gorav Jindal

[1]Department of Computer Science, Saarland University

[2]Department of Computer Science, Aalto University

January 12, 2019
ITCS 2019

Symmetric Boolean Functions

Definition

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is said to be symmetric if it is invariant under any permutation of its inputs.

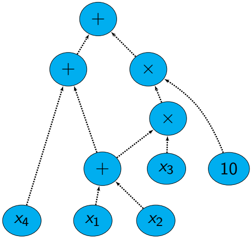
- Can a symmetric Boolean function be hard to compute?

Fact

A symmetric Boolean function only depends on the number of 1's in the input and thus can be computed by constant depth threshold circuits (complexity class TC^0*). Therefore "easy" to compute.*

Polynomials and Arithmetic Circuits

- Every arithmetic circuit computes a polynomial and vice versa.



- This circuit computes the polynomial $F \in \mathbb{C}[x_1, x_2, x_3, x_4]$ where $F = 10x_3(x_1 + x_2) + x_1 + x_2 + x_4$.
  - Size and depth have same definitions as in the Boolean case.

Arithmetic Complexity

Definition

The arithmetic complexity $L(f)$ of a polynomial $f \in \mathbb{C}[x_1, x_2, \ldots, x_n]$ is defined as the minimum size of any arithmetic circuit computing F.

- Thus $L(F) \leq 10$, where $F = 10x_3(x_1 + x_2) + x_1 + x_2 + x_4$.

Hard Polynomial families

Goal

Find polynomial families $\{f_1, f_2, \ldots, f_n, \ldots, \}$ such that $L(f_n)$ is a super polynomial function of n.

- The permanent family defined by $\mathrm{per}_n \stackrel{\mathrm{def}}{=\!=} \sum_{\pi \in \mathfrak{S}_n} \prod_{i=1}^{n} x_{i,\pi(i)}$ is believed to be "hard".
  - Known as VP vs VNP conjecture.

Symmetric Polynomials

Definition

A polynomial $f \in \mathbb{C}[x_1, x_2, \ldots, x_n]$ is said to be symmetric if it is invariant under any permutation of its inputs.

Example

$x_1^2 + x_2^2 + x_1 x_2 \in \mathbb{C}[x_1, x_2]$ is symmetric whereas $x_1^2 + x_2$ is not.

Question

Lipton and Regan (Gödel's Lost Letter and P $=$ NP, 2009) ask whether we can find hard (families of) symmetric polynomials?

Elementary Symmetric Polynomials

Definition

The i^{th} elementary symmetric polynomial e_i in n variables $x_1, x_2, \ldots, x_n$ is defined as:

$$e_i \stackrel{\text{def}}{=\!=} \sum_{1 \leq j_1 < j_2 < \cdots < j_i \leq n} x_{j_1} \cdot x_{j_2} \cdot \cdots \cdot x_{j_i}.$$

- e_i's are obviously symmetric.
- Sum and product of symmetric polynomials is also symmetric.
- Thus the polynomials in the algebra generated by e_i's are also symmetric.

Fundamental Theorem

Theorem (Fundamental Theorem of Symmetric Polynomials)

If $g \in \mathbb{C}[x_1, x_2, \ldots, x_n]$ is a symmetric polynomial, then there exists a unique polynomial $f \in \mathbb{C}[y_1, y_2, \ldots, y_n]$ such that $g = f(e_1, e_2, \ldots, e_n)$.

- Write symmetric polynomials always with f_{Sym}. Thus we have the isomorphism $f(e_1, e_2, \ldots, e_n) = f_{\mathrm{Sym}}$:

$$f \iff f_{\mathrm{Sym}}.$$

Idea

Study the connection between $L(f)$ and $L(f_{\mathrm{Sym}})$.

Relation between $L(f)$ and $L(f_{\mathrm{Sym}})$

Lemma

For all $f \in \mathbb{C}[x_1, x_2, \ldots, x_n]$, $L(f_{\mathrm{Sym}}) \leq L(f) + O(n^2)$.

Proof.

Replace x_i by e_i, e_i's can be computed a circuit of size $O(n^2)$. □

- Can we also bound $L(f)$ polynomially in terms of $L(f_{\mathrm{Sym}})$? This is what Lipton and Regan (Gödel's Lost Letter and $\mathrm{P} = \mathrm{NP}$, 2009) ask.
- If we can bound $L(f)$ polynomially in terms of $L(f_{\mathrm{Sym}})$, then we get that for a "hard" polynomial f, f_{Sym} is also hard.

Main Theorem

Theorem

For any polynomial $f \in \mathbb{C}[x_1, x_2, \ldots, x_n]$ of degree d,
$L(f) \leq \tilde{O}\left(d^2 L(f_{\text{Sym}}) + d^2 n^2\right)$.

- Previous best bound: $L(f) \leq 4^n (n!)^2 (L(f_{\text{Sym}}) + 2)$.

Corollary

Assuming VP $\neq$ VNP, *symmetric polynomial family* $(q_n)_{n \in \mathbb{N}}$ *defined by* $q_n \stackrel{def}{=\!=} (\text{per}_n)_{\text{Sym}}$ *has super polynomial arithmetic complexity.*

Main idea

Example

Suppose $f_{\text{Sym}} = x_1^2 + x_2^2 + x_1x_2 = e_1^2 - e_2$. Given an arithmetic circuit for f_{Sym}, we want to get a circuit for $f = e_1^2 - e_2$.

Idea

x_1, x_2 are the roots of polynomial
$B(y) \stackrel{\text{def}}{=\!=} y^2 - (x_1 + x_2)y + x_1x_2 = y^2 - e_1y + e_2$. Thus:

$$x_1 = \frac{e_1 + \sqrt{e_1^2 - 4e_2}}{2}. \quad (1)$$

$$x_2 = \frac{e_1 - \sqrt{e_1^2 - 4e_2}}{2}. \quad (2)$$

Main idea(Continued)

- If we substitute:

$$x_1 = \frac{e_1 + \sqrt{e_1^2 - 4e_2}}{2}. \tag{3}$$

$$x_2 = \frac{e_1 - \sqrt{e_1^2 - 4e_2}}{2}. \tag{4}$$

in the circuit for f_{Sym}, we obtain a circuit for f. How to compute the above radical expressions?

- These are not even polynomials.

Main idea(Continued)

- Use the substitution $e_2 \leftarrow e_2 - 1$ and then substitute x_1 and x_2 in $f_{\text{Sym}}(x_1, x_2)$ to obtain $f(e_1, e_2 - 1)$.
  - But even after this $e_2 \leftarrow e_2 - 1$, radical expressions for x_1, x_2 are not polynomials.
- But they are power series (use Taylor expansion).
  - We can not compute power series using arithmetic circuits.

Idea

Only need to compute degree two truncations of these power series, because f is of degree two.

Example

- $x_1 = \frac{e_1}{2} + \sqrt{1+E}$, where $E = \frac{e_1^2}{4} - e_2$.
- $\sqrt{1+E} = 1 + \frac{E}{2} - \frac{E^2}{8} + \cdots + =$
  $1 + \frac{1}{2}\left(\frac{e_1^2}{4} - e_2\right) - \frac{1}{8}\left(\frac{e_1^2}{4} - e_2\right)^2 + \cdots + .$
- The degree ≤ 2 part of $\sqrt{1+E}$ is $1 + \frac{1}{2}\left(\frac{e_1^2}{4} - e_2\right) - \frac{1}{8}e_2^2$.

Example (Continued)

- Substitute $x_1 = \frac{e_1}{2} + 1 + \frac{1}{2}\left(\frac{e_1^2}{4} - e_2\right) - \frac{1}{8}e_2^2$ and $x_2 = -\frac{e_1}{2} + 1 + \frac{1}{2}\left(\frac{e_1^2}{4} - e_2\right) - \frac{1}{8}e_2^2$ in $f_{\mathrm{Sym}} = x_1^2 + x_2^2 + x_1x_2$.
- After substitution, $f = e_1^2 - (e_2 - 1) +$ terms of degree atleast 3.
- Junk terms can be removed efficiently.
- Now use the substitution $e_2 \leftarrow e_2 + 1$ to obtain $f = e_1^2 - e_2$.

General idea

- Consider $B(y, e_1, e_2, \ldots, e_n) = y^n - e_1 y^{n-1} + \ldots + (-1)^n e_n$.
- $x_1, x_2, \ldots, x_n$ are the roots of $B(y)$.
- Use the substitution $e_n \leftarrow e_n + (-1)^{n-1}$ as in the case of $n = 2$.

Problem

Abel-Ruffini theorem states that for $n > 4$, roots of $B(y)$ are not radicals in $e_1, e_2, \ldots, e_n$ ($\mathfrak{S}_n$ is not solvable for $n > 4$).

Computing Roots of $B(y)$

- Roots of $B(y)$ are not radicals in $e_1, e_2, \ldots, e_n$.

Fact

Roots of $B(y)$ are power series in $e_1, e_2, \ldots, e_n$ as in the case of $n = 2$.

- Low degree truncation of these roots can be computed by using the Newton iteration.
- Again, junk terms can be removed efficiently.

Final algorithm

Problem

Given a circuit C_{Sym} computing a symmetric polynomial f_{Sym}, find a circuit computing f with $\deg(f) = d$.

- Consider:

$$B(y, e_1, e_2, \ldots, e_n) = y^n - e_1 y^{n-1} + \ldots + (-1)^n(e_n + (-1)^{n-1}).$$

- Compute roots of $B(y)$ up-till degree d by using the Newton iteration.
- Remove Junk terms (terms of degree $> d$). Use the substitution $e_n \leftarrow e_n - (-1)^{n-1}$ to obtain a circuit for f.

Thanks

Thank you for your attention!