1. Algebraic Circuits

Friday, August 11, 2023 4:34 PM

We study complexity problems about (multivariate) polynomials over a field $\mathbb{F}$.

Field: $\mathbb{C}, \mathbb{R}, \mathbb{Q}$, where you can add, subtract, and multiply elements. And if $a \neq 0$, then you can divide by a.

Another example: $\mathbb{F}_p = \{0, 1, \dots, p-1\}$, where the outcome of arithmetic operations is taken modulo p.

For simplicity, you may assume $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$.

Def: A monomial in variables $X_1, \dots, X_n$ is a formal product $X_1^{e_1} \cdots X_n^{e_n}$ where $e_1, \dots, e_n \in \mathbb{N}$. Note: $X_i X_j = X_j X_i$.

A polynomial in $X_1, \dots, X_n$ over a field $\mathbb{F}$ is a linear combination of monomials in $X_1, \dots, X_n$ over $\mathbb{F}$, i.e. it has the form

$$f(X_1, \dots, X_n) = \sum_{i \in I} c_i X_1^{e_{i1}} \cdots X_n^{e_{in}},$$

where I is a finite set and $c_i \in \mathbb{F}$.

Write $\mathbb{F}[X_1, \dots, X_n]$ for the polynomial ring, consisting of all polynomials in $X_1, \dots, X_n$ over $\mathbb{F}$.

How fast can we compute a polynomial, in terms of # of arithmetic operations needed?

Example: Let $M = \begin{pmatrix} X_{11} & \cdots & X_{1n} \\ X_{21} & \cdots & X_{2n} \\ \vdots & \ddots & \vdots \\ X_{n1} & \cdots & X_{nn} \end{pmatrix}$. Then $\det(M)$ is a polynomial in $X_{11}, \dots, X_{nn}$.

$$\mathrm{Det} := \det(M) = \sum_{\substack{\text{permutation} \\ \sigma \text{ of } \{1, \dots, n\}}} \mathrm{sgn}(\sigma) \prod_{i=1}^{n} X_{i\sigma(i)} \quad \text{where } \mathrm{sgn}(\sigma) = \pm 1.$$

It has $n!$ monomials.

So trivially, Det can be computed using $n! \cdot O(n)$ operations.

Is this the best we can do?

Is this the best we can do?

Def: An algebraic circuit C (a.k.a. arithmetic circuit) over $\mathbb{F}$ is a directed acyclic graph (DAG) where each node (gate) has one of the following 4 labels:

1. $+$ (addition)
2. $\times$ (multiplication)

} non-input gates

3. a variable x_i
4. a constant $c \in \mathbb{F}$

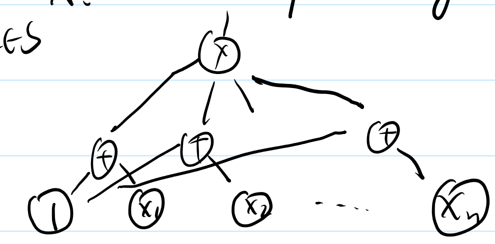
} input gates

size of C = # of gates (sometimes # of wires)

depth of C = length of the longest path.

Note that C in variables $x_1, \cdots, x_n$ computes a polynomial in $\mathbb{F}[x_1, \cdots, x_n]$. We use C to denote both the circuit and the polynomial it computes.

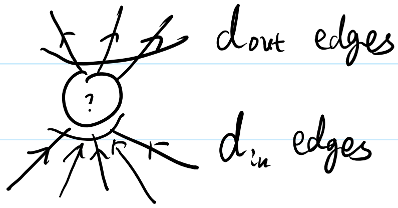
Example: $\sum_{S \subseteq [n]} \prod_{i \in S} x_i$ is computed by the circuit



size $= 2n+2$

depth $= 2$.

for a gate:



We say the gate has fan-in d_{in} and fan-out d_{out}.

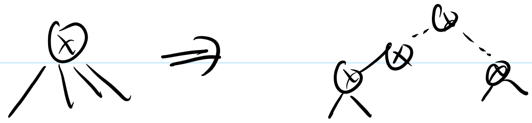
fan-in / fan-out of the circuit is the maximum fan-in / fan-out over all gates.

Fanin: Two most interesting fan-ins: 2 or unbounded

The above example has unbounded fan-in, as $\times$ has n in-edges.

The above example has unbounded fan-in, as $\otimes$ has n in-edges.

It can be turned into a fan-in 2 circuit:



depth: $2 \to O(\log n)$

We assume fan-in is 2 from now on, unless noticed otherwise.

Fan-out: An algebraic circuit is an algebraic formula if fan-out $= 1$.

i.e., Cannot reuse gates.

Some Complexity classes:

$VP = VP_{\mathbb{F}} =$ set of polynomials $f \in \mathbb{F}[x_1, \cdots, x_n]$ of poly(n) degree computed by poly(n)-sized circuits.

(More precisely, it is the set of polynomial families $(f_i)_{i \in \mathbb{N}}$)

Usually drop the subscript $\mathbb{F}$. (You may assume $\mathbb{F} = \mathbb{C}$).

VF (a.k.a. VP_e) $=$ set of polynomials f of poly(n) degree computed by poly(n)-sized formulas.

We will see $\mathrm{Det} \in VP$.

$VNP :=$ set of $f(x_1, \cdots, x_n)$ of poly(n) degree such that

$\exists\, g(x_1, \cdots, x_n, y_1, \cdots, y_m) \in VP$. $m \le$ poly(n),

and $$f(x_1, \cdots, x_n) = \sum_{(y_1, \cdots, y_m) \in \{0,1\}^m} g(x_1, \cdots, x_n, y_1, \cdots, y_m).$$

Obviously $VP \subseteq VNP$.

(Valiant's conjecture) $VP \neq VNP$.

$$\mathrm{Perm} := \sum_{\sigma \in S_n} \prod_{i=1}^{n} x_{i\sigma(i)} \qquad \left(\text{Recall } \mathrm{Det} = \sum_{\sigma \in S_n} \mathrm{sgn}(\sigma) \prod_{i=1}^{n} x_{i\sigma(i)}\right)$$

We will show $\mathrm{Perm} \in VNP$, and is VNP-complete in some sense.

$\sigma \in S_n \quad i=1$

We will show Perm $\in$ VNP, and is VNP-complete in some sense.

It is conjectured Perm $\notin$ VP.

VNC^k = set of f of poly(n) degree, computed by poly(n)-sized & $(\log(n))^k$-depth circuits

$VNC = \bigcup_{k=0}^{\infty} VNC \subseteq VP$.

In Boolean complexity, $NC^1 \subseteq \cdots NC^k \subseteq \cdots \subseteq NC \subseteq P$

And it is conjectured $NC \neq P$, i.e., not all poly-time algorithms are parallelizable

However, perhaps surprisingly, $VNC^2 = VP$!

and $VNC^1 = VF$

$VF = VNC^1 \subseteq VNC^2 = VP$.

Why study algebraic complexity?

1. clean, no need to worry $+, \times$ on bit level.
2. Valiant argued it is easier to prove lower bound in algebraic models

an algebraic circuit in VP computing $f \Rightarrow$ boolean circuit in P/poly computing f

by "simulating" the arithmetic operations.

So lower bound for P/poly $\Rightarrow$ lower bound for VP

i.e. the latter is easier.

However, for more restricted models, this $\Rightarrow$ may not hold.

Ex. parity is not in AC^0, constant-depth unbounded fan-in boolean circuits.

In fact it admits an exponential lower bound against AC^0.

Only recently, Limaye-Srinivasan-Tavenas

proved a superpolynomial lower bound for algebraic circuits of constant depth and unbounded fan-in.

algebraic circuits of constant depth and unbounded fan-in.

(FOCS'21 best paper)

3. algebraic tools may be useful.

In the definition of algebraic circuits we have $+$ and $\times$, but not $-$ and $/$. Why?

$a-b$ can be computed as $a+(-1)\cdot b$.

a/b ?

<u>Thm</u>: Allowing division does not make VP more powerful.

Similar result holds for VF.

Will show this next time.