Towards Multilinear Depth-3 PIT

- A depth-3 circuit is the following polynomial representation:

$\Sigma\Pi\Sigma \rightarrow$
$$C(x_1,\dots,x_n) = \sum_{i\in[k]} \prod_{j\in[d]} \ell_{ij}$$

where ℓ_{ij} is a linear polynomial in $\mathbb{F}[x_1,\dots,x_n]$.

- The size of the circuit is ndk.
+ bits required to represent the constants in $\mathbb{F}$.

- Depth-3 PIT Qn: Check $C(\bar{x}) \stackrel{?}{=} 0$ in poly(size(C)) time.

- PIT question can be asked for any arithmetic circuit model.

▷ It has a poly-time randomized algorithm.

- In fact, the randomized algorithm suggests the existence of a poly-sized $\mathcal{H}_s \subset \mathbb{F}^n$ s.t. $\forall$ circuits C of size s:

$$C \neq 0 \Rightarrow \exists \alpha \in \mathcal{H}_s,\ C(\alpha) \neq 0.$$

▷ For circuit families "small" hitting-sets exist.

- Blackbox PIT Qn: Can a hitting-set family be constructed in det. poly-time?

- We are mainly interested in blackbox PIT algorithms.

(In contrast to whitebox PIT.)

- This question is related to circuit lower bound questions.

PIT ⇒ LBs : [HS'80, KI'04, Agr'05, ...]

depth3 ⇒ general: [AV'08, Koi'12, Tav'13, GKKS'13]

- In the last decade, several results have been shown for depth-3 covering both — lower bounds & PIT.

  But they are with restrictions.

- In this talk we restrict to multilinear depth-3.

  I.e. $C = \sum_i \prod_j \ell_{ij}$, where $\forall i$,

  $\{\ell_{i1}, \ldots, \ell_{id}\}$ are supported on disjoint variables.

- Eg. $x_1^2 x_2$ cannot be computed by multilr. depth3. While $(1+x_1)\cdots(1+x_n)$ can.

- For this model:
  - exponential (2^n) lower bounds by [RY'09].
  - sub-exponential ($n^{n^{2/3} \cdot \lg n}$) blackbox PIT by [de Oliveira, Shpilka, Volk '15].

- We begin the study with set-multilinear depth-3.

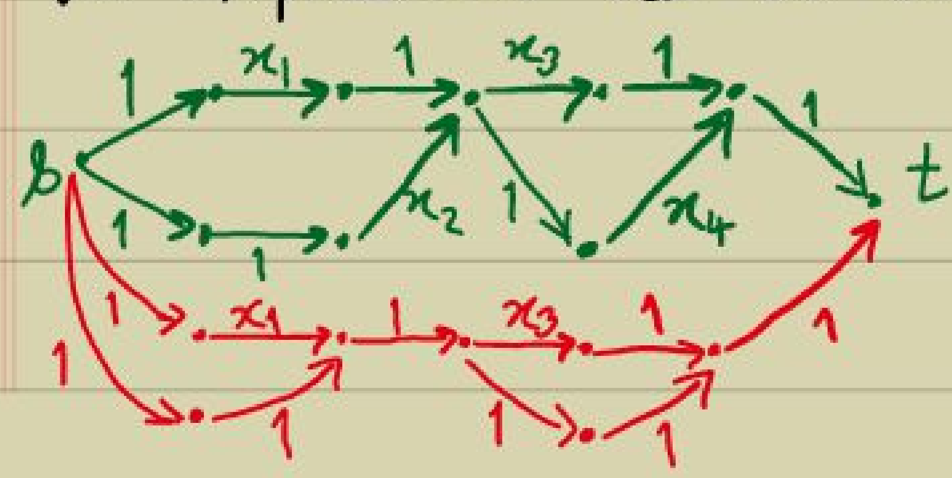
  I.e. $\forall i$, $\{l_{i1}, \ldots, l_{id}\}$ induces the same partition on the variables $[n]$.

- Eg. $(1+x_1)\ldots(1+x_n)+1$ is set-multilinear, while $(x_1+x_2)(x_3+x_4)+(x_1+x_3)(x_2+x_4)$ is not (syntactically).

- Set-multilinear depth-3 have a special ABP (arithmetic branching program):

  all variables separated
  (or read-once oblivious).

- Eg. $(x_1+x_2)(x_3+x_4)+(1+x_1)(1+x_3) =: C(\bar{x})$ can be expressed as a branching prog.:



of length $\approx n$
width $\approx kn$

- Further, we get the matrix product :

$$C(\bar{x}) = \bar{1}^T \cdot \begin{pmatrix} x_1 & & & \\ & 1 & & \\ & & x_1 & \\ & & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & & & \\ x_2 & 0 & & \\ & & 1 & \\ & & & 0 \end{pmatrix} \cdot \begin{pmatrix} x_3 & 1 & & \\ & 0 & & \\ & & x_3 & 1 \\ & & & 0 \end{pmatrix} \cdot$$

$$\begin{pmatrix} 1 & & & \\ x_4 & 0 & & \\ & & 1 & \\ & & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} .$$

- We call such a matrix product $\prod_{i=1}^{n} A_i(x_i) =: D(\bar{x}) \in M_w(\mathbb{F})[\bar{x}]$ an ROABP,
& it computes the polynomial $C(\bar{x}) = \bar{c}^T \cdot D(\bar{x}) \cdot \bar{d}$ for $\bar{c}, \bar{d} \in \mathbb{F}^w$.
(C is said to have a width-w ROABP.)

- PIT algorithms are known (almost):
1) whitebox poly(nw)-time [RS'05]
2) blackbox $w^{O(\lg n)}$-time [AGKS'15].

open: Remove $\lg n$?

- (1) is based on evaluation dimension / coeff. space of D.
(2) " " " rank concentration in D.

Idea of (1)

- Given $D = A_1(x_1) \cdots A_n(x_n) \in M_w(\mathbb{F})[\bar{x}]$
  we want to test $C = \bar{c}^T \cdot D \cdot \bar{d} \stackrel{?}{=} 0$.

- Algo. sketch:
  (i) For $i = 2$ to n
  (ii) expand $D_i = D_{i-1} \cdot A_i(x_i)$
  (iii) if sparsity$(D_i) > w^2$ then
  coefficients in D_i are $\mathbb{F}$-dependent.
  Keep an $\mathbb{F}$-basis & drop the extra monomials.
  (What remains is called D_i.)
  (iv) Test whether $\bar{c}^T \cdot D_n \cdot \bar{d} \stackrel{?}{=} 0$.

▷ coeff-span$_{\mathbb{F}}(D)$ = coeff-span$_{\mathbb{F}}(D_n)$ &
sparsity$(D_n) \leq w^2$.

▷ This takes time poly(nw).
But is whitebox!

Idea of (2)

- We are given only an oracle to $C(\bar{x}) = \bar{c}^T \cdot D \cdot \bar{d}$, where $D = \prod_i A_i(x_i)$.

- The idea is to find a map $\phi : x_i \mapsto t^{w_i}$ s.t. a least basis of coeff-span$_{\mathbb{F}}(D)$ gets isolated in $\phi(D)$.

$|S| \leq w^2 \rightarrow$ I.e. $\exists$ monomials $S \subseteq$ support(D) s.t.

(i) $\text{span}_{\mathbb{F}}\{\text{coeff}(m)(D) \mid m \in S\} = \text{coeff-span}_{\mathbb{F}}(D)$,

& (ii) $\forall m' \notin S$, $\text{coeff}(m')(D) \in \text{span}_{\mathbb{F}}\{\text{coeff}(m)(D) \mid m \in S, \phi(m) < \phi(m')\}$.

▷ If ϕ isolates a least basis in D then: $C \neq 0 \Rightarrow \phi(C) \neq 0$.

- Thus, all we need to do is to construct "small" weights w_i s.t. ϕ isolates a least basis.

- The idea for ϕ is to attempt a recursion on the ROABP length.
(Giving a $\lg n$ in the exponent.)

<u>Sketch</u>: • Say, the two halves $L = \prod_{i=1}^{n/2} A_i$ & $R = \prod_{i>n/2} A_i$

have a least basis isolated under a monomial map
$\psi : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{F}[t_1, \ldots, t_e]$.

$\Rightarrow \psi(L) = \underline{\text{least-basis-part}} + \text{rest}$
& $\psi(R) = \underline{\text{least-basis-part}'} + \text{rest}'$

• Note that by extending ψ (s.t. the w^t monomials in the product of the two least basis parts are preserved) we achieve least basis isolation in $L \cdot R$!

- This extension of φ needs a new variable $t_{\ell+1}$ with the ordering $t_1 < t_2 < \dots < t_{\ell+1}$ & individual degrees $\mathrm{poly}(w)$.

$\Rightarrow$ By using this recursive step on contiguous blocks of $2, 2^2, \dots, 2^{\lg n}$, we get the map φ in time $w^{O(\lg n)}$.
(We get $w^{O(\lg n)}$ candidate φ, each $\lg n$-variate & individual-degree $\mathrm{poly}(w)$.)

- Recently, [GKST'15] have extended this to a sum of constantly many ROABP's.
e.g. multilinear depth-3 with few underlying partitions.