Bootstrapping variables in circuits

Nitin Saxena (CSE@IIT Kanpur, India)

(Joint work with Manindra Agrawal & Sumanta Ghosh, STOC'18)

2018, Université Paris Diderot

Contents

- Polynomial identity testing
- Hardness/ de-randomness & a conjecture
- Partial Hsg
- Perfect Bootstrapping
- Shallow Bootstrapping
- Constant Bootstrapping
- Conclusion

Polynomial identity testing

- Given an arithmetic circuit $C(x_1, \ldots, x_n)$ of size s, whether it is zero?
  - In $\text{poly}(s)$ many bit operations?
  - Think of field $F =$ finite field, rationals, numberfield, or localfield.
- Brute-force expansion is as expensive as s^s.
- Randomization gives a practical solution.
  - Evaluate $C(x_1, \ldots, x_n)$ at a random point in F^n.
  - (Ore 1922), (DeMillo & Lipton 1978), (Zippel 1979), (Schwartz 1980).
- This test is blackbox, i.e. one does not need to see C.
  - Whitebox PIT – where we are allowed to look inside C.
- Blackbox PIT is equivalent to designing a hitting-set $H \subset F^n$.
  - H contains a non-root of *each* nonzero $C(x_1, \ldots, x_n)$ of size s.

Polynomial identity testing

- Question of interest: Design hitting-sets for circuits.
  - Appears in numerous guises in computation.
- Complexity results
  - Interactive protocol (Babai,Lund,Fortnow,Karloff,Nisan,Shamir 1990), PCP theorem (Arora,Safra,Lund,Motwani,Sudan,Szegedy 1998), …
- Algorithms
  - Graph matching, matrix completion (Lovász 1979), equivalence of branching programs (Blum, et al 1980), interpolation (Clausen, et al 1991), primality (Agrawal,Kayal,S. 2002), learning (Klivans, Shpilka 2006), polynomial root testing (Kopparty, Yekhanin 2008), factoring (Shpilka, Volkovich 2010 & Kopparty, Saraf, Shpilka 2014), alg.independence test (Pandey, S. ,Sinhababu, 2016), approx.root finding (Guo, S. ,Sinhababu, 2018), ….

Polynomial identity testing

- Hitting-sets relate to circuit lower bounds.
- It is conjectured that VP≠VNP. (Valiant's Hypothesis 1979)
  - Or, permanent is *harder* than determinant?
- "proving permanent hardness" flips to "designing hitting-sets".
  - *Almost,* (Heintz,Schnorr 1980), (Kabanets,Impagliazzo 2004), (Agrawal 2005 2006), (Dvir,Shpilka,Yehudayoff 2009), (Koiran 2011) ...
- Designing an efficient algorithm leads to awesome tools!
- Connections to *Geometric Complexity Theory* and derandomizing the *Noether's normalization lemma*. (Mulmuley 2011, 2012, 2017)

Hitting-set generator (Hsg)

- *Functional* version of hitting-set $H \subset F^n$ for polynomials $\mathcal{P}$:
  - Consider $f(y) := (f_1(y), \ldots, f_n(y))$ whose evaluations contain H.
- Call $f(y)$ a (t,d)-hsg for family $\mathcal{P}$ if the $f_i(y)$'s are time-t computable and have degree $\leq d$.
  - By t-hsg or time-t blackbox PIT we mean a (t,t)-hsg.
- A poly(s)-degree hsg for size-s circuits can be designed in PSPACE.
  - Hint: the hsg exists and verified via Hilbert's Nullstellensatz.
- (Mulmuley 2012, 2017) What about poly(s)-degree hsg for $\overline{\text{VP}}$?
  - Designable in PSPACE as well! (Guo, S. ,Sinhababu, 2018)

Contents

- Polynomial identity testing
- Hardness/ de-randomness & a conjecture
- Partial Hsg
- Perfect Bootstrapping
- Shallow Bootstrapping
- Constant Bootstrapping
- Conclusion

A Working Conjecture

- Pseudorandomness in boolean circuits:
  - (Nisan,Wigderson 1994) Optimal prg for P/poly exists iff E-computable $2^{\Omega(n)}$-hard function family exists.
- Could we prove:
  - Poly-time hsg for VP exists iff E-computable $2^{\Omega(n)}$-hard polynomial family exists ?
- **Conjecture-LB:** E-computable $2^{\Omega(n)}$-hard polynomial family exists.
  - This family $\{f_n\}_n$ has individual-degree (ideg) *constant*.
  - $\text{Coeff}(x^e)(f_n)$ is $2^{O(n)}$-computable.
- Implies: Either $E \not\subseteq \#P/poly$ OR VNP is $2^{\Omega(n)}$-hard.

Hsg gives Conjecture-LB-- Annihilator

- (Heintz, Schnorr 1980) essentially showed that a poly-time hsg implies Conjecture-LB.
  - *Idea:* If $f(y)= (f_1(y), \ldots, f_n(y))$ is an hsg for size-s degree-s circuits $\mathcal{P}_s$,
    then consider a *nonzero* annihilator $A(z_1, \ldots, z_{\log s})$ such that $A(f_1(y), \ldots, f_{\log s}(y))=0$.
  - A is E-computable, by linear algebra.
  - A is not in $\mathcal{P}_s$. Thus, $A(z_1, \ldots, z_m)$ is $s^{\Omega(1)}=2^{\Omega(m)}$-hard.
  - *Note:* 1) A exists with ideg constant.
  - 2) The proof only uses the hsg on the first log-variables!

Conjecture-LB "gives" Hsg-- NW Design

- (Kabanets,Impagliazzo 2004) essentially showed that Conjecture-LB implies a *quasi*poly-time hsg.
  - *Idea:* Let q_m be an E-computable $2^{\Omega(m)}$-hard polynomial family.
  - Let P be a nonzero size-s degree-s circuit.
  - Define $\ell := c_2 \log s > m := c_1 \log s$.
  - *Nisan-Wigderson Design:* Stretch the few variables $z_1, \ldots, z_\ell$ to the s polynomials $q_m(T_1), \ldots, q_m(T_s)$, where T_i's are almost disjoint m-sets.
  - Suppose $P(q_m(T_1), \ldots, q_m(T_s))$ vanishes. Then, by circuit factoring (Kaltofen 1989) q_m has a *small* circuit. Contradiction!
  - We get a poly-time $s \mapsto O(\log s)$ variable reduction for VP. □

Contents

- Polynomial identity testing
- Hardness/ de-randomness & a conjecture
- Partial Hsg
- Perfect Bootstrapping
- Shallow Bootstrapping
- Constant Bootstrapping
- Conclusion

Partial Hsg

- Prior proof ideas suggest that even *partial* hsg is of interest.
  - Significantly smaller variate circuits.
- Let $\mathbf{g_{s,m}} = (g_{s,1}(y), \ldots, g_{s,m}(y))$ be hsg for size-s degree-s circuits $\mathcal{P}_s$ that depend only on first m variables.
- If $m=s^{1/c}$ then the partial hsg *gives* a complete hsg for $\mathcal{P}_s$.
  - Blow up size $s \mapsto s^c$.
- If $m=s^{o(1)}$ then the partial hsg seems *weak*.
  - Naively, a size blow up of $s \mapsto s^{\omega(1)}$.
  - i.e. *super-poly* blow up to get a complete hsg.

Partial Hsg-- Bootstrap question

- Bootstrap hsg: For $m=s^{o(1)}$, given a ``small'' $\mathbf{g}_{s,m}$ could you devise a ``small'' $\mathbf{g}_{s,s}$?
- What about $m= \log\log s$?
- $m= \log^{oc} s$? $m= \log^{\star} s$?
- $m= 6913$? $m= 3$?
- YES! (*In this work*)
- Bootstrapping means that we only need to study extremely low-variate circuits.
  - To prove Conjecture-LB.

Contents

- Polynomial identity testing
- Hardness/ de-randomness & a conjecture
- Partial Hsg
- Perfect Bootstrapping
- Shallow Bootstrapping
- Constant Bootstrapping
- Conclusion

Perfect Bootstrapping

- Let's start with a partial hsg for a *tiny* $n = \omega(\log\log s)$.
  - Let $f(y) = (f_1(y), \ldots, f_n(y))$ be s^e-hsg for size-s deg-s n-variate circuits $\mathcal{P}_{s,2}$.

- Bootstrap in *three* main steps:

- 1) Partial hsg to hard polynomial.
  - Fix $m := c_1 \log\log s$.
  - Consider a *nonzero* annihilator $A(z_1, \ldots, z_m)$ such that $A(f_1(y), \ldots, f_m(y)) = 0$. Denote A by $q_{m,s}$.
  - $q_{m,s}$ is $\mathrm{poly}(s)$-time computable, by linear algebra.
  - $q_{m,s}$ is not in $\mathcal{P}_{s,2}$. Thus, $q_{m,s}$ is s-hard.
  - *Note*- ideg of $q_{m,s}$ is $s^{3e/m}$, so is non-constant. □

Perfect Bootstrapping-- Step 2

- 2) Hard polynomial to Variable reduction.
  - Define $s' := s^{c_0}$, $\ell := c_2 \text{loglog } s' > m' := c_1 \text{loglog } s'$ and $N := 2^{\text{loglog } s'} \approx \log s$.
  - Let P be a nonzero size-s degree-s N-variate circuit.
  - We want to *stretch* the few variables $z_1, \ldots, z_\ell$ to N polynomials $q_{m',s'}(T_1), \ldots, q_{m',s'}(T_N)$, where T_i's are almost disjoint m'-sets. (*NW-design*)
  - Suppose $P(q_{m',s'}(T_1), \ldots, q_{m',s'}(T_N))$ vanishes. Then, by circuit factoring (Kaltofen 1989) $q_{m',s'}$ has a *small* circuit. Contradiction!
  - We get a poly-time ($\log s \mapsto O(\text{loglog } s)$) variable reduction for VP. □

Perfect Bootstrapping-- Step 3

- 3) Reusing the partial hsg.
  - Recall $s' := s^{c_0}$, $\ell := c_2 \text{loglog } s' > m' := c_1 \text{loglog } s'$ and $N := 2^{\text{loglog } s'} \approx \log s$.
  - Let P be a *nonzero* size-s degree-s N-variate circuit.
  - $P(q_{m',s'}(T_1),..., q_{m',s'}(T_N)) \neq 0$.
  - It involves the few variables $z_1, \ldots, z_\ell$.
  - So, use the s^e-hsg known for circuits $\mathcal{P}_{s,2}$. □

- Repeating this shows: Partial hsg for *tiny* $m = \omega(\text{loglog } s)$ gives the complete hsg in deterministic poly-time.

- **Theorem:** Partial hsg for $m = \log^{oc} s$ yields complete hsg in deterministic poly-time.
  - Any constant c.

Contents

- Polynomial identity testing
- Hardness/ de-randomness & a conjecture
- Partial Hsg
- Perfect Bootstrapping
- Shallow Bootstrapping
- Constant Bootstrapping
- Conclusion

Shallow Bootstrapping

- Let's start with a partial hsg for depth-4 with a *tiny* $n \geq 3$.
  - Let $f(y) = (f_1(y), \ldots, f_n(y))$ be $(\text{poly}(s^n), O(s^{n/2}/\log^2 s))$-hsg for size-$s$ deg-s n-variate depth-4 circuits $\mathcal{P}_s$.

- Get a partial hsg for multilinear polynomials computed by depth-4 with $m := n \log s$ variables.
  - Form n blocks of $\log s$ variables each.
  - Apply n disjoint Kronecker maps locally ($x_i \mapsto y^{2^i}$). Size grows to s^2 and nonzeroness preserved.

- Let $g(y) = (g_1(y), \ldots, g_m(y))$ be $(\text{poly}(s^n), O(s^n/\log^2 s))$-hsg for degree $m/2$ multilinear polynomials $\mathcal{P}'_s$ computed by size-s m-variate depth-4 circuits.

Shallow Bootstrapping-- Step 1

- Bootstrap in *two* main steps:

- 1) Partial hsg to hard polynomial.
  - Recall: $\mathcal{P}'_s$ is multilinear, deg $m/2$ and $m=n\log s$ variate.
  - Consider a *nonzero* annihilator $A(z_1, \ldots, z_m)$ such that $A(g_1(y), \ldots, g_m(y))=0$. Denote A by q_m .
  - q_m is poly(s)-time computable, by linear algebra.
  - q_m is not in $\mathcal{P}'_s$. Thus, q_m is s-hard *for depth-4*.
  - *Note*- We can find q_m multilinear & deg $m/2$, as:
  - #monomials $> 2^m/\sqrt{(2m)} > O(s^n/\log^2 s).m >$ #constraints.

  - By (Agrawal,Vinay 2008), q_m is $s=2^{\Omega(m/n)}$ -hard *for VP*. □

Shallow Bootstrapping-- Step 2

- 2) Hard polynomial to Variable reduction.
  - Note- q_m is an E-computable $2^{\Omega(m)}$-hard polynomial family.
  - As seen before, using *NW-design* & *circuit factoring*, we get:
  - A poly-time $s \mapsto O(\log s)$ variable reduction for VP. □
- After variable reduction, we can trivially design $s^{O(\log s)}$-hsg.
- **Theorem:** $(\text{poly}(s^n), O(s^{n/2}/\log^2 s))$-hsg for size-$s$ n-variate depth-4 circuits yields quasi-hsg for VP.
  - Any constant $n \geq 3$ works!
  - Trivial is $(\text{poly}(s^n), (s+1)^n)$-hsg.
  - $\Sigma\Lambda\Sigma\Pi$ or $\Sigma\Pi\Sigma\Lambda$ circuits suffice.
  - Poly-hsg for log-variate *ΣΠΣ circuits*/ *width-2-ABP* suffices too!

Contents

- Polynomial identity testing
- Hardness/ de-randomness & a conjecture
- Partial Hsg
- Perfect Bootstrapping
- Shallow Bootstrapping
- Constant Bootstrapping
- Conclusion

Constant Bootstrapping

- Let $m_0 < f_0$ be constants.
- Let $g(y) = (g_1(y), \ldots, g_{m0}(y))$ be $O(s^{f0})$-hsg for size-s deg-s m_0-variate circuits $\mathcal{P}_{s,0}$.
- *NW design*: $(\ell := m_0 , m_0/8f_0 , d := m_0/16f_0^2)$ and $m_1 := 2\hat{}(d/4)$.
- Bootstrap in *three* main steps:
- 1) Partial hsg for $\mathcal{P}_{s,0}$ to hard polynomial.
  - $q_{0,s}$ is $m_0/8f_0$ variate.
  - $q_{0,s}$ is s^{4f0}-time computable, by linear algebra.
  - $q_{0,s}$ is not in $\mathcal{P}_{s,0}$. Thus, $q_{0,s}$ is s-hard.
  - ideg of $q_{0,s}$ is $\approx s\hat{}(8f_0^2/m_0)$, so is non-constant. □

Constant Bootstrapping-- Step 2

- 2) Hard polynomial to Variable reduction.
  - Define s':= s^7 and $m_1 = 2^{(m_0/64f_0^2)}$.
  - Let P be a nonzero size-s degree-s m_1-variate circuit.
  - We want to *stretch* the few variables $z_1, \ldots, z_\ell$ to m_1 polynomials $q_{0,s'}(T_1),\ldots, q_{0,s'}(T_{m1})$,
    where T_i's are almost disjoint $(m_0/8f_0)$-sets. (*NW-design*)
  - Suppose $P(q_{0,s'}(T_1),\ldots, q_{0,s'}(T_{m1}))$ vanishes. Then, by circuit factoring (Kaltofen 1989) $q_{0,s'}$ has *size*<*s*' circuit. Contradiction!
  - We get $\approx s^{(f_0 \log f_0)}$ -time $(m_1 \mapsto m_0)$ variable reduction for size-s deg-s m_1-variate circuits $\mathcal{P}_{s,1}$. □

Constant Bootstrapping-- Step 3

- 3) Reusing the partial hsg.
  - Recall s'= s^7, $\ell = m_0$ and $m_1 = 2^{(m_0/64f_0^2)}$.
  - Let P be a *nonzero* size-s degree-s m_1-variate circuit.
  - $P(q_{0,s'}(T_1),..., q_{0,s'}(T_{m1})) \neq 0$.
  - It involves the few variables $z_1, \ldots, z_\ell$.
  - So, use the appropriate $O(s^{f0})$-hsg known for circuits $\mathcal{P}_{s,0}$.
  - Overall, it takes time $O(s^{(16f_0^2)})$.
  - So, we define $f_1 := 16f_0^2$. □
- After i repetitions, we get $O(s^{f_i})$-hsg for size-s deg-s m_i-variate circuits $\mathcal{P}_{s,i}$.
  - Thus, hsg for constant-variate circuits can be bootstrapped. □

Constant Bootstrapping

- For a rapid completion we need $m_1 = 2\hat{}(m_0/64f_0^2) \gg 2\hat{}(m_0^{1-\varepsilon})$, for a constant $\varepsilon>0$.
  - **Tetration** ensures completion in $O(\log^\star s)$ iterations.
- **Theorem 1:** $O(s^2)$-hsg for $m=6913$ yields complete hsg in deterministic $s^{\exp\exp(O(\log^* s))}$-time.
  - Trivial is $O(s^{6913})$-hsg.
- *Note*-- We need m_0 slightly larger than f_0^2 .
- **Theorem 2:** For constant $\delta<1/2$, $s^{n\hat{}\delta}$-hsg for size-s degree-s n-variate circuits yields $s^{\exp\exp(O(\log^* s))}$-time hsg for size-s degree-s circuits.
  - Trivial is $O(s^n)$-hsg.
  - Actually, $(O(s^n), s^{n\hat{}\delta})$-hsg will suffice!

Contents

- Polynomial identity testing
- Hardness/ de-randomness & a conjecture
- Partial Hsg
- Perfect Bootstrapping
- Shallow Bootstrapping
- Constant Bootstrapping
- Conclusion

At the end …

- *Powerful* bootstrapping of partial hsg for width-2 ABP, depth-3, depth-4 and VP models.
- Each of these partial hsg imply Conjecture-LB.
  - Could we connect *directly* to VP≠?VNP ?
- Could we design any of these partial hsg (nontrivially)?
- Design $(s^{2\hat{}n}, s^{n/2})$ -hsg for size-s ΣΠΣ(n) ?
- Blackbox PIT for $O(\log^{\star} s).\log s$ -variate size-s diagonal depth-3 circuits.
  - (Forbes,Ghosh,S. 2018) solved size-s ΣΛΣ(log s) case.

Thank you!