Lecture 20 Sums-of-Squares

Nov 10

First: when I defined SA Boolean axioms were encoded as two inequalities

$$\cancel{\{x_i^2 - x_i \geq 0,\ x_i - x_i^2 \geq 0\}} \rightarrow \{x_i^2 - x_i = 0\}$$

Multilinearize $x_1^2\, x_2^2\, x_3^2 \underset{\text{prove}}{\rightsquigarrow} x_1 x_2 x_3$

$$x_1^{100}\, x_2^{100}\, x_3^{100} + x_2^2 x_3^2 (x_1 - x_1^2) + x_1 x_3^2 (x_2 - x_2^2) + x_1 x_2 (x_3 - x_3^2) = x_1 x_2 x_3$$

These are not non-neg juntas!

Instead we should always include $x_i^2 - x_i = 0$ instead of two inequalities, or just multilinearize with every computation (working modulo $\{x_i^2 - x_i = 0\}$).

On to SOS!

Recall the SA set-up: let $P = \{p_1 = 0, \cdots, p_m = 0\}$ (include $x_i^2 - x_i$ for all i) and $Q = \{q_1 \geq 0, \cdots, q_\ell \geq 0\}$.

An SA proof of the inequality $r(x) \geq c$ is given by

- List of polynomials $f_1 \cdots f_m$

- List of conical juntas $J_1, \cdots, J_\ell$

such that

$$\sum_{i=1}^{m} f_i p_i + \sum_{j=1}^{\ell} J_j q_j = r(x) - c$$

Why not allow any non-negative poly here?

- We need to be able to verify that this is a proof.

<u>Fact</u> Testing if $g(x_1, \cdots, x_n) \geq 0$ for all $x \in \{0,1\}^n$ is coNP-Hard.

- In SoS we replace non-negative juntas with a different family of non-negative polys:

Sum-of-Squares

<u>Defn</u> A polynomial $g(x)$ is a sum-of-squares if we can represent

$$g = \sum_i h_i^2$$

for some polynomials $\{h_i\}$.

<u>Defn</u> Let P, Q be as above. An SoS-derivation of $r(x) \geq c$ is given by

- A list of polynomials $f_1 \cdots f_m$
- A list of SoS polynomials $g_1 \cdots g_\ell$

such that

$$\sum f_i p_i + \sum g_j q_j = r(x) - c.$$

- The degree of the derivation is the max degree of any polynomial in it.
- The size is the length of the encoding of the proof in bits.

→ coefficient size really matters!

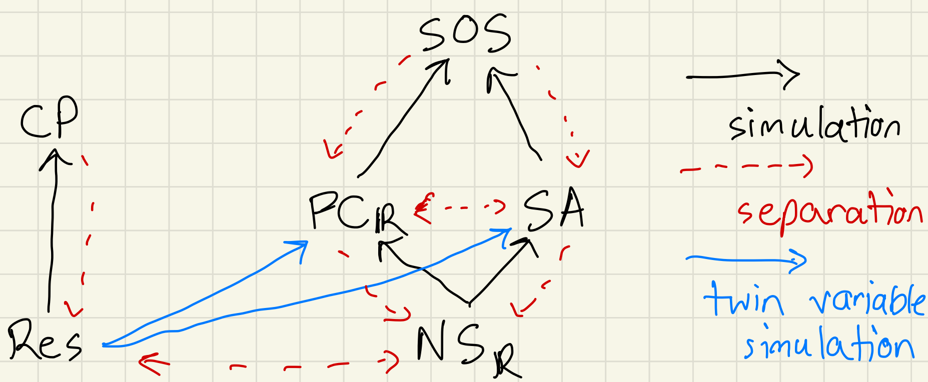
Q1. Can we represent any non-negative poly as an SOS?

NO! [Hilbert 1888] Non-constructive

[Motzkin 1967] "Motzkin polynomial"

$$p(x,y) = x^4y^2 + x^2y^4 - 3x^2y^2 + 1$$

How does SOS compare to other proof systems?



– [Berkholz 18] SOS can simulate $PC_{\mathbb{R}}$ efficiently if boolean axioms are present

(Not true without boolean axioms)

<u>Theorem</u> There are degree-3 SOS refut. of PHP^{n+1}_{n}.

<u>Pf.</u> Recall the encoding:

$$\forall i \in [n+1]: \quad \sum_{j=1}^{n} x_{ij} \geq 1$$

$$\forall i \neq j \in [n+1],\ k \in [n] \qquad x_{ik} + x_{jk} \leq 1 \qquad\qquad 1 \geq 0$$

We derive for all $k \in [n]$:

$$\sum_{i=1}^{n+1} x_{ik} \leq 1$$

$$\sum_{\substack{i,j \in [n+1] \\ i \neq j}} (1 - x_{ik} - x_{jk})\, x_{ik}^2 \quad + \quad 1 \cdot \left(1 - \sum_{i=1}^{n+1} x_{ik}\right)^2$$

$$\left(1 - \sum_{i=1}^{n+1} x_{ik}\right)^2 = 1 - 2\sum_{i=1}^{n+1} x_{ik} + \left(\sum_{i=1}^{n+1} x_{ik}\right)^2$$

|| multilinearize

$$1 - \cancel{2} \sum_{i=1}^{n+1} x_{ik} \cancel{+ \sum_{i=1}^{n+1} x_{ik}} + \sum_{i \neq j} x_{ik} x_{jk}$$

$$\sum_{\substack{i,j \in [n+1] \\ i \neq j}} (1 - x_{ik} - x_{jk})\, x_{ik}^2 = \sum_{i \neq j} x_{ik}^2 - x_{ik}^3 - x_{ik}^2 x_{jk}$$

$$\underset{\text{multilin}}{=} \sum_{i \neq j} - x_{ik} x_{jk} \quad + \quad \left(1 - \sum_{i=1}^{n+1} x_{ik} + \sum_{i \neq j} x_{ik} x_{jk}\right)$$

$$= 1 - \sum_{i=1}^{n+1} x_{ik}$$

Then summing up $\forall k$ $\left(\sum_{i=1}^{n+1} x_{ik}\right) - 1$ and $1 - \sum_{i=1}^{n+1} x_{ik}$ we get -1. $\square$

For both Tseitin and random 3-CNF formulas, $\Omega(n)$-degree lower bounds are known for SOS. [Grigoriev 01, Schoenbeck 08]

<u>Theorem</u> [Atserias–Hakoniemi 19]

If there is a size-s SOS refutation of a system (P, Q) then

\# of monomials

$$\deg_{SOS}((P;Q)) \leq O(\sqrt{n \log s} + k)$$

where k is the degree of the system.

Open Problem Give strong separations between CP and SOS.

Open Problem Prove that SOS is **not** size automatizable under any reasonable complexity assumption.

SOS is degree-automatizable (like SA): if there is a degree $\leq d$ SOS proof of some inequality then there is an algorithm that will find it in time $n^{O(d)}$ *

using **semidefinite programming** (which generalizes LPs).

For many NP-Hard problems (Max Cut, Vertex Cover, CSPs) SOS-based semidefinite programming algorithms capture the best-known approximation algs.

Furthermore: assuming the Unique Games Conjecture the approximation ratio obtained by SOS for these problems is optimal unless $P = NP$.

ex) <u>Max-Cut</u>: Given a graph $G = (V, E)$ with edge weights $w_e \geq 0$.

Find a partition $V = V_1 \cup V_2$ such that the total weight of edges crossing the partition is maximized.

- NP-Complete [Karp]
- There is a simple LP obtains a $\frac{1}{2}$-approximation ratio [Folklore]
- [de la Vega-Mathieu 07 Charikar-Makarychev^2 09 O'donnell-Schramm 18] SA requires degree $\Omega(n)$ to obtain $\left(\frac{1}{2}+\varepsilon\right)$-approximation
- There is an SDP algorithm captured by $O(1)$ SOS obtains a ~ 0.878-approximation [Goemans-Williamson 94]

$$0.878\ldots = \min_{0<\theta<\pi} \frac{\theta/\pi}{(1-\cos\theta)/2}$$

This is tight (unless $P=NP$) assuming UGC
[Kindler-Khot-Mossel-O'Donnell 04]

UGC It is NP-Hard to solve the $(1-\varepsilon, \varepsilon)$-Unique Games problem.

$(1-\varepsilon, \varepsilon)$
Unique Games Given graph $G=(V,E)$ and for each edge given a permutation function

$$\mu_e : [q] \to [q]$$

Goal: Find a colouring $x : V \to [q]$ s.t. for every edge uv

$$\mu_e(x(u)) = x(v)$$

s.t either $\geq (1-\varepsilon)$ frac of const are sat or

$\leq \varepsilon$-frac of constraints are satisfied.