Duality of sum of nonnegative circuit polynomials and optimal SONC bounds

Dávid Papp

MEGA 2021 (Tromsø/virtual). June 7–11, 2021.

NC STATE UNIVERSITY

Supported in part by NSF DMS-1719828 and NSF DMS-1847865.

Nonnegativity certificates

Let $p \in \mathbb{R}[x_1, \ldots, x_n]_d$ be a polynomial and $S \subseteq \mathbb{R}^n$.

Definition (nonnegativity certificate)

A certificate of nonnegativity (usually) is a representation of p that makes its nonnegativity over S apparent.

- Sum-of-squares (SOS)
- *Sum-of-nonnegative-circuit-polynomials (SONC)*
- etc.
- Different: dual certificates (see Maria Macaulay's talk)

Optimization = certified upper bound (a point $\mathbf{x} \in S$) + certified lower bound (nonnegativity certificate for some $p - c$).

SONC polynomials

- Polynomials certifiably nonnegative by the AM/GM inequality. (“agiforms”, [Reznick 1989])
- Systematically/computationally studied by de Wolff et al. [2016–]
- Related: Ghasemi and Marshall; Chandrasekaran and Shah (SAGE)

Basic idea (example):

- the Motzkin polynomial $m(x, y) = x^4y^2 + x^2y^4 - 3x^2y^2 + 1$ is not SOS but is nonnegative,
- because each monomial is nonnegative and $\frac{x^4y^2+x^2y^4+1}{3} \geq x^2y^2$.
- Easily generalized to weighted AM/GM.

SONC polynomials

- Polynomials certifiably nonnegative by the AM/GM inequality. (“agiforms”, [Reznick 1989])
- Systematically/computationally studied by de Wolff et al. [2016–]
- Related: Ghasemi and Marshall; Chandrasekaran and Shah (SAGE)

Basic idea (example):

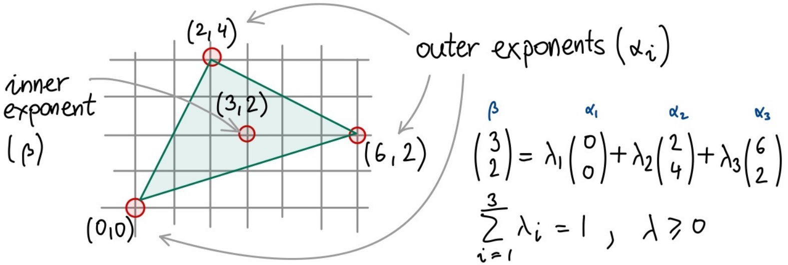
- the Motzkin polynomial $m(x,y) = x^4y^2 + x^2y^4 - 3x^2y^2 + 1$ is not SOS but is nonnegative,
- because each monomial is nonnegative and $\frac{x^4y^2+x^2y^4+1}{3} \geq x^2y^2$.
- Easily generalized to weighted AM/GM.

SONC polynomials

Definition (circuit polynomial)

An n-variate polynomial p is a circuit polynomial if it is given by $p(\mathbf{x}) = \sum_{i=1}^{r} p_{\alpha_i}\mathbf{x}^{\alpha_i} + p_{\beta}\mathbf{x}^{\beta}$, where

- $\operatorname{supp}(p) = \{\alpha_1, \ldots, \alpha_r, \beta\}$ is minimally affinely dependent and
- $\beta \in \operatorname{conv}(\{\alpha_1, \ldots, \alpha_r\})$.



SONC polynomials

Proposition (Iliman & de Wolff, 2016)

Let p be an n-variate circuit polynomial satisfying $p(\mathbf{x}) = \sum_{i=1}^{r} p_{\boldsymbol{\alpha}_i}\mathbf{x}^{\boldsymbol{\alpha}_i} + p_{\boldsymbol{\beta}}\mathbf{x}^{\boldsymbol{\beta}}$ for some real coefficients $p_{\boldsymbol{\alpha}_i}$ and $p_{\boldsymbol{\beta}}$ and suppose that $\boldsymbol{\beta} = \sum_{i=1}^{r} \lambda_i \boldsymbol{\alpha}_i$ with some $\lambda_i > 0$ satisfying $\sum_{i=1}^{r} \lambda_i = 1$.
Then p is nonnegative if and only if $\boldsymbol{\alpha}_i \in (2\mathbb{N})^n$ and $p_{\boldsymbol{\alpha}_i} > 0$ for each i, and at least one of the following two alternatives holds:

1. $\boldsymbol{\beta} \in (2\mathbb{N})^n$ *and* $p_{\boldsymbol{\beta}} \geq 0$, *or*
2. $|p_{\boldsymbol{\beta}}| \leq \prod_{i=1}^{r} \left(\frac{p_{\boldsymbol{\alpha}_i}}{\lambda_i}\right)^{\lambda_i}$.

- Makes the recognition of a nonnegative circuit polynomial straightforward.
- SONC = Minkowski sum of nonnegative circuit polynomials with different supports.
- Optimal SONC bounds and certificates?

$$\inf\{p(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^n\} = \sup\{c \mid p-c \text{ is nonnegative}\} \geq \sup\{c \mid p-c \text{ is SONC}\}.$$

SONC polynomials and convex conic representations

$$|p_{\beta}| \leq \prod_{i=1}^{r} \left(\frac{p_{\alpha_i}}{\lambda_i}\right)^{\lambda_i} \quad (\boldsymbol{\lambda} > 0,\ \sum_{i=1}^{r} \lambda_i = 1)$$

- This inequality is convex in the coefficients of p.
    - (Dressler et al. 2017): it can be represented using $O(r)$ relative entropy and linear constraints.
    - (Wang & Magron, 2020): it can be represented using second-order cone constraints.
    - Lifted SOCP representation also follows from (Alizadeh & Goldfarb, 2003).
- In this talk: treat it as a single convex (dual power) cone constraint (no lifting).

SONC polynomials and power cones

Definition (the power cone and its dual)

The *(generalized) power cone* with *signature* $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_r) \in (0,1)^r$ is the convex cone defined as

$$\mathcal{P}_{\boldsymbol{\lambda}} \stackrel{\text{def}}{=} \left\{ (\mathbf{v}, w) \in \mathbb{R}^r_+ \times \mathbb{R} \;\middle|\; |w| \leq \prod_{i=1}^{r} v_i^{\lambda_i} \right\}.$$

The dual of this (closed, pointed, full-dimensional) convex cone is

$$\mathcal{P}^*_{\boldsymbol{\lambda}} \stackrel{\text{def}}{=} \left\{ (\mathbf{v}, w) \in \mathbb{R}^r_+ \times \mathbb{R} \;\middle|\; |w| \leq \prod_{i=1}^{r} \left(\frac{v_i}{\lambda_i}\right)^{\lambda_i} \right\}.$$

In other words, $|p_{\boldsymbol{\beta}}| \leq \prod_{i=1}^r \left(\frac{p_{\boldsymbol{\alpha}_i}}{\lambda_i}\right)^{\lambda_i} \iff ((p_{\boldsymbol{\alpha}_1}, \ldots, p_{\boldsymbol{\alpha}_r}), p_{\boldsymbol{\beta}}) \in \mathcal{P}^*_{\boldsymbol{\lambda}}$.

- Note that the cone depends on the circuit $\text{supp}(p) = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_r, \boldsymbol{\beta}\}$ only through its $\boldsymbol{\lambda}$ vector.

Theoretical consequences

The conic formulation allows for simple (convex) analysis on the SONC cone.

Theorem (P., 2019)

Every SONC polynomial p has a SONC decomposition in which every nonnegative circuit polynomial is supported on a subset of $\mathrm{supp}(p)$.

- A generalization of (Wang, 2019).
- Proof is an elementary application of convex programming duality.

Support of SONC decompositions

- Fix a set of circuits $\mathcal{C} = \{C^1, \ldots, C^N\}$ covering supp(p).
- Let $\mathcal{S}(\mathcal{C})$ be the set of SONC polynomials corresponding to $\mathcal{C}$.
- Let V be the vertices of New(p), and consider the optimization problem

$$\begin{aligned} \underset{\boldsymbol{\gamma}\in\mathbb{R}_+^V}{\text{minimize}} \quad & \sum_{\boldsymbol{\alpha}\in V} \gamma_{\boldsymbol{\alpha}} \\ \text{subject to} \quad & \Big(\mathbf{x} \mapsto p(\mathbf{x}) + \sum_{\boldsymbol{\alpha}\in V} \gamma_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}\Big) \in \mathcal{S}(\mathcal{C}). \end{aligned} \tag{P}$$

- Its dual can be written as

$$\begin{aligned} \underset{\mathbf{y}\in\mathbb{R}^{\text{supp}(p)}}{\text{maximize}} \quad & -\mathbf{p}^{\mathrm{T}}\mathbf{y} \\ \text{subject to} \quad & (y_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}\in C^j} \in \mathcal{P}_{\boldsymbol{\lambda}(C^j)} \quad j = 1, \ldots, N \\ & y_{\boldsymbol{\alpha}} \geq 0 \qquad \boldsymbol{\alpha} \in \text{supp}(p) \cap (2\mathbb{N})^n, \\ & y_{\boldsymbol{\alpha}} \leq 1 \qquad \boldsymbol{\alpha} \in V. \end{aligned} \tag{D}$$

Support of SONC decompositions

- Both the primal and dual problem have optimal solutions, and strong duality holds (min $=$ max).
- Consider two instances of these primal-dual pairs:
  1. where $\mathcal{C}$ is the set of all circuits $\subseteq \text{supp}(p)$, and
  2. where $\mathcal{C}$ is the set of all circuits $\subseteq \text{New}(p) \cap \mathbb{N}^n$.
- The optimal solutions of the first pair are optimal solutions of the second pair, proving that $f \in \mathcal{S}(\mathcal{C})$ holds either for both set of circuits or for neither.
- The proof of the last statement uses linear(!) programming to argue that the extra power cone constraints are redundant.

Computational consequences

- To find the optimal SONC bound (using all possible circuits), we need to solve
$$\inf\{\gamma \mid p + \gamma \in \mathcal{S}(\mathcal{C})\}.$$
- The previous theorem helps reduce $\mathcal{C}$.
- The number of circuits (and dual power cone constraints) is still exponential!
- Various heuristics have been proposed for a selection of "good" circuits. (Seidler, de Wolff, Dressler, etc.)
- The proof suggests a different approach:
  - Iteratively identify and add the "useful" circuits using the dual problem.
  - Stop with a certificate that no circuits can improve the bound.

Computational consequences

Algorithm 1: Optimal SONC bound with iterative circuit generation

input : A polynomial p.
outputs: The optimal SONC lower bound for p and a SONC decomposition certifying the bound.

```
1  initialize 𝒞 = {C^1, ..., C^N}
2  repeat
3      solve (P)-(D) for the optimal SONC bound γ* and dual vector y*
4      find circuits C ⊆ supp(p) whose power cone constraint is violated
5      if no such circuit exists then
6          return γ* and the SONC decomposition of p + γ*
7      else
8          add the circuit(s) C found in Step 4 to 𝒞
9      end if
10 until false
```

Step 4 amounts to solving $< |\text{supp}(p)|$ small linear programs. (No enumeration.)

Example

Consider the polynomial p given by

$$p(x_1, x_2) = 1 + x_2^2 - x_1^2 x_2^2 + x_1^2 x_2^6 + x_1^6 x_2^2.$$

- p clearly has a SONC lower bound that we can get by taking $\mathcal{C} = \{C_1\} = \{\{(0,0), (2,6), (6,2), (2,2)\}\}$.
- Solving the (P)-(D) problems, we obtain $\gamma^* = -\frac{7}{8}$ and the SONC decomposition

$$p(x_1, x_2) - \frac{7}{8} = (x_2)^2 + \underbrace{\left(\frac{1}{8} + x_1^2 x_2^6 + x_1^6 x_2^2 - x_1^2 x_2^2\right)}_{nonneg.\ circuit\ poly.}.$$

- The dual optimal solution is $\mathbf{y}^* = (1, 0, \frac{1}{4}, \frac{1}{16}, \frac{1}{16})$.
- Circuit generation: (next slide)

Example cont'd

- Circuit generation:
  1. No circuit with $(0,2)$ as an inner exponent can violate its corresponding power cone constraint.
  2. $C_2 = \{(0,2),(6,2),(2,2)\}$, with signature $\boldsymbol{\lambda}(C_2) = \left(\frac{2}{3}, \frac{1}{3}\right)$ violates its power cone constraint.
- Reoptimize with $\mathcal{C} = \{C_1, C_2\}$: $\gamma^* = -1$, the new SONC decomposition is

$$p(x_1, x_2) - 1 = x_1^2 x_2^6 + \underbrace{\left(x_2^2 + x_1^6 x_2^2 - x_1^2 x_2^2\right)}_{nonneg.\ circuit\ poly(C_2)}.$$

- The new dual solution is $\mathbf{y}^* = (1,0,0,0,0)$.
- Circuit generation: no circuits violate their power cone constraint.
- This is the optimal SONC bound. (In this case, 1 is also the global minimum.)

Implementation

Some practical consideration for the implementation:

- We can solve the (P)-(D) instances in each iteration using an SOCP (following Wang & Magron), which allows for an "off-the-shelf" implementation.
  - Reconstructing the optimal dual vector from the lifted dual vector isn't obvious, though!
- We can also optimize directly over Cartesian products of power cones (and their duals) using interior-point methods of non-symmetric cone programming.
- We used `alfonso` (open source, Matlab) for the conic programs.
  - Avoids the lifting necessary to represent the problems as an SOCP.
  - Efficient: same complexity of solving linear programs of the same size.

Numerical experiments

Questions of interest:

1. How many circuit generation iterations are needed?
2. How many circuits until the algorithm stops?

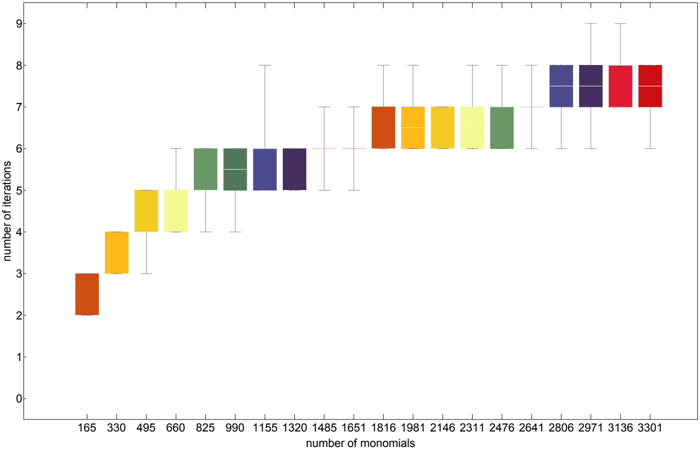
Used two sets of test problems:

1. The largest problems of (Seidler & de Wolff, 2018) with general (not simplex) Newton polytopes.
   (438 instances; $4 \leq n \leq 40$, $6 \leq d \leq 60$; sparse.)
2. Random instances with $n = 25, d = 8$, all exponents even, with varying $|\operatorname{supp}(p)|$ up to the maximum.

Numerical experiments

How many circuit generation iterations are needed?

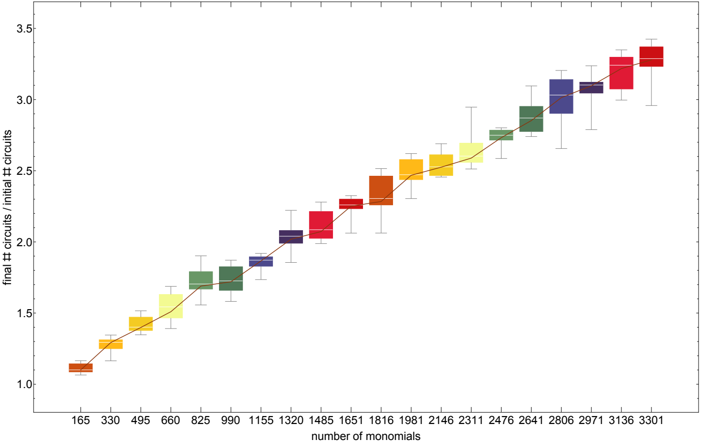
- Appears clearly sublinear in $|\text{supp}(p)|$.
- The largest number encountered was 9.



Numerical experiments

How many circuits until the algorithm stops?

- The final/initial set of circuits appears to increase linearly with $|\text{supp}(p)|$

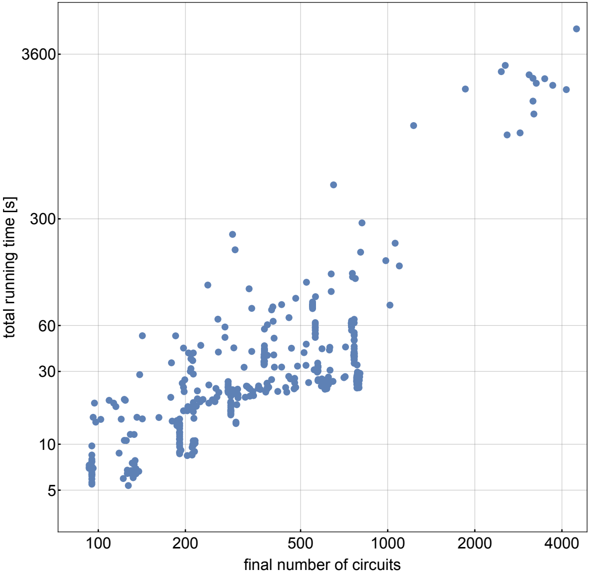


Summary and references

- Optimal SONC bounds can be computed quickly without resorting to approximations or circuit selection heuristics.
- The circuit generation algorithm identifies the “right” circuits efficiently.
- For a given set of circuits, the SONC decomposition can be computed using non-symmetric cone optimization, circumventing costly characterizations of the SONC cone as the projection of “simpler” high-dimensional cones.

- D.P.: Duality of sum of nonnegative circuit polynomials and optimal SONC bounds. `https://arxiv.org/abs/1912.04718`
- D.P. and S. Yıldız: alfonso: Matlab package for nonsymmetric conic optimization. *INFORMS Journal on Computing*. `https://arxiv.org/abs/2101.04274`
- Matlab code: `https://github.com/dpapp-github/alfonso`

Running time / number of circuits (Seidler–de Wolff)



Running time / number of circuits (larger instances)