Sums of Squares with Multipliers: Advantages and Limitations

Greg Blekherman [1]   João Gouveia [2]   James Pfeiffer [3]

[1]Georgia Tech

[2]Universidade de Coimbra

[3]University of Washington

October 1st - CWMINLP 2013 - Paris

Sums of Squares

Checking if a polynomial is a sum of squares (sos) is easy

Sums of Squares

Checking if a polynomial is a sum of squares (sos) is easy

SOS verification

Let $\deg(p(x)) = 2d$ and $\bar{x}$ be the vector of monomials of degree up to d. $p(x)$ is a sos iff there is $A \succeq 0$ such that $p(x) = \bar{x}^t A \bar{x}$.

Sums of Squares

Checking if a polynomial is a sum of squares (sos) is easy

SOS verification

Let $\deg(p(x)) = 2d$ and $\bar{x}$ be the vector of monomials of degree up to d. $p(x)$ is a sos iff there is $A \succeq 0$ such that $p(x) = \bar{x}^t A \bar{x}$.

Why?

Sums of Squares

Checking if a polynomial is a sum of squares (sos) is easy

SOS verification

Let $\deg(p(x)) = 2d$ and $\bar{x}$ be the vector of monomials of degree up to d. $p(x)$ is a sos iff there is $A \succeq 0$ such that $p(x) = \bar{x}^t A \bar{x}$.

Why?

$$p(x) = \sum h_i(x)^2$$

Sums of Squares

Checking if a polynomial is a sum of squares (sos) is easy

SOS verification

Let $\deg(p(x)) = 2d$ and $\bar{x}$ be the vector of monomials of degree up to d. $p(x)$ is a sos iff there is $A \succeq 0$ such that $p(x) = \bar{x}^t A \bar{x}$.

Why?

$$p(x) = \sum h_i(x)^2 = \sum_i \left\langle \hat{h}_i, \bar{x} \right\rangle^2$$

Sums of Squares

Checking if a polynomial is a sum of squares (sos) is easy

SOS verification

Let $\deg(p(x)) = 2d$ and $\bar{x}$ be the vector of monomials of degree up to d. $p(x)$ is a sos iff there is $A \succeq 0$ such that $p(x) = \bar{x}^t A \bar{x}$.

Why?

$$p(x) = \sum h_i(x)^2 = \sum_i \left\langle \hat{h}_i, \bar{x} \right\rangle^2 = \bar{x}^t \left(\sum \hat{h}_i \hat{h}_i^t \right) \bar{x}$$

Sums of Squares

Checking if a polynomial is a sum of squares (sos) is easy

SOS verification

Let $\deg(p(x)) = 2d$ and $\bar{x}$ be the vector of monomials of degree up to d. $p(x)$ is a sos iff there is $A \succeq 0$ such that $p(x) = \bar{x}^t A \bar{x}$.

Why?

$$p(x) = \sum h_i(x)^2 = \sum_i \left\langle \hat{h}_i, \bar{x} \right\rangle^2 = \bar{x}^t \left(\sum \hat{h}_i \hat{h}_i^t \right) \bar{x}$$

Demanding a polynomial to be sos is a semidefinite constrain in the coefficients.

Sums of Squares in global polynomial optimization

Global Polynomial Optimization

$p_{\min} = \min_{x \in \mathbb{R}^n} p(x)$

Sums of Squares in global polynomial optimization

Global Polynomial Optimization

$p_{\min} = \min_{x \in \mathbb{R}^n} p(x) = \max \lambda$ s.t. $p(x) - \lambda$ is nonnegative.

Sums of Squares in global polynomial optimization

Global Polynomial Optimization

$p_{\min} = \min_{x \in \mathbb{R}^n} p(x) = \max \lambda$ s.t. $p(x) - \lambda$ is nonnegative.

Deciding nonnegativity of a polynomial is hard, hence we relax it.

Global Polynomial Optimization Relaxation

$p_{\text{sos}} = \max \lambda$ s.t. $p(x) - \lambda$ is sos.

Sums of Squares in global polynomial optimization

Global Polynomial Optimization

$p_{\min} = \min_{x \in \mathbb{R}^n} p(x) = \max \lambda$ s.t. $p(x) - \lambda$ is nonnegative.

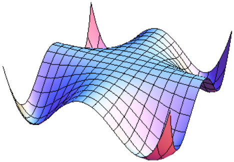
Deciding nonnegativity of a polynomial is hard, hence we relax it.

Global Polynomial Optimization Relaxation

$p_{\text{sos}} = \max \lambda$ s.t. $p(x) - \lambda$ is sos.

It does not always work.

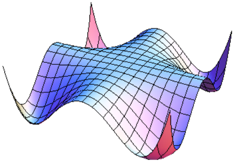
Hierarchy of Sums of Squares



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

For Motzkin polytope $p_{\text{min}} = 0$ but $p_{\text{sos}} = +\infty$.

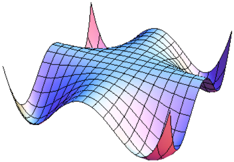
Hierarchy of Sums of Squares



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

For Motzkin polytope $p_{\min} = 0$ but $p_{\text{sos}} = +\infty$.
However $(x^2 + y^2)^2 M(x, y)$ is a sum of squares, which is enough to guarantee nonnegativity.

Hierarchy of Sums of Squares



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

For Motzkin polytope $p_{\text{min}} = 0$ but $p_{\text{sos}} = +\infty$.
However $(x^2 + y^2)^2 M(x, y)$ is a sum of squares, which is enough to guarantee nonnegativity.

This motivates a new hierarchy:

Global Polynomial Optimization Relaxation Hierarchy

$p_{\text{sos},k} = \max \lambda$ s.t. $\|x\|^{2k}(p(x) - \lambda)$ is sos.

Properties of this hierarchy

Good News

For all polynomials, $p_{\text{sos},k} \to p_{\text{min}}$.

Properties of this hierarchy

Good News

For all polynomials, $p_{\text{sos},k} \to p_{\min}$.

Bad News

For some polynomials, $p_{\text{sos},k} \neq p_{\min}$ for any k.

Properties of this hierarchy

Good News

For all polynomials, $p_{\text{sos},k} \to p_{\min}$.

Bad News

For some polynomials, $p_{\text{sos},k} \neq p_{\min}$ for any k.

$$D(x, y, z, w) = x^4y^2w^2 + y^4z^2w^2 + x^2z^4w^2 - 3x^2y^2z^2w^2 + z^8$$

For $D_{\text{sos},k} > 0$ for all k.

Properties of this hierarchy

Good News

For all polynomials, $p_{\text{sos},k} \to p_{\min}$.

Bad News

For some polynomials, $p_{\text{sos},k} \neq p_{\min}$ for any k.

$$D(x, y, z, w) = x^4y^2w^2 + y^4z^2w^2 + x^2z^4w^2 - 3x^2y^2z^2w^2 + z^8$$

For $D_{\text{sos},k} > 0$ for all k.

However $(x^2 + y^2 + z^2)D(x, y, z, w)$ is sos, which is enough.

Free multiplier Hierarchy

Picking the right multiplier definitely helps.

Free multiplier Hierarchy

Picking the right multiplier definitely helps. So we might as well search all of them:

Global Polynomial Optimization Relaxation Hierarchy v2.0

$p^*_{\text{sos},k} = \max \lambda$ s.t. $q(x)(p(x) - \lambda)$ is sos, and $q(x) \neq 0$ is sos.

Free multiplier Hierarchy

Picking the right multiplier definitely helps. So we might as well search all of them:

Global Polynomial Optimization Relaxation Hierarchy v2.0

$p^*_{\text{sos},k} = \max \lambda$ s.t. $q(x)(p(x) - \lambda)$ is sos, and $q(x) \neq 0$ is sos.

Good News

For every polynomial there exists k such that $p_{\min} = p^*_{\text{sos},k}$.

Free multiplier Hierarchy

Picking the right multiplier definitely helps. So we might as well search all of them:

Global Polynomial Optimization Relaxation Hierarchy v2.0

$p^*_{\text{sos},k} = \max \lambda$ s.t. $q(x)(p(x) - \lambda)$ is sos, and $q(x) \neq 0$ is sos.

Good News

For every polynomial there exists k such that $p_{\min} = p^*_{\text{sos},k}$.

Bad News

Not an SDP anymore (not convex)

Free multiplier Hierarchy

Picking the right multiplier definitely helps. So we might as well search all of them:

Global Polynomial Optimization Relaxation Hierarchy v2.0

$p^*_{\text{sos},k} = \max \lambda$ s.t. $q(x)(p(x) - \lambda)$ is sos, and $q(x) \neq 0$ is sos.

Good News

For every polynomial there exists k such that $p_{\text{min}} = p^*_{\text{sos},k}$.

Bad News

Not an SDP anymore (not convex)

Not so Bad News

It is however a quasi-convex problem, hence still doable. It is also OK for fixed λ.

Copositive Matrices

Cone of Copositive Matrices

$\text{CoP}_n = \{M \in \mathbb{R}^{n \times n} \ : \ M = M^t, \ x^t M x \geq 0, \ \forall x \geq 0\}.$

Copositive Matrices

Cone of Copositive Matrices

$\text{CoP}_n = \{M \in \mathbb{R}^{n \times n} \ : \ M = M^t, \ x^t M x \geq 0, \ \forall x \geq 0\}.$

Copositive programming is an elegant and efficient way of stating hard problems.

Copositive Matrices

Cone of Copositive Matrices

$$\mathrm{CoP}_n = \{M \in \mathbb{R}^{n\times n} \ : \ M = M^t, \ x^t M x \geq 0, \ \forall x \geq 0\}.$$

Copositive programming is an elegant and efficient way of stating hard problems.

Checking copositivity is very hard.

Copositive Matrices

Cone of Copositive Matrices

$\text{CoP}_n = \{M \in \mathbb{R}^{n \times n} \;:\; M = M^t,\; x^t M x \geq 0,\; \forall x \geq 0\}$.

Copositive programming is an elegant and efficient way of stating hard problems.

Checking copositivity is very hard.

Simple Copositivity Criteria

$\text{PSD}_n + \text{NN}_n \subseteq \text{CoP}_n$.

Copositive Matrices

Cone of Copositive Matrices

$\text{CoP}_n = \{M \in \mathbb{R}^{n\times n} \ : \ M = M^t, \ x^t M x \geq 0, \ \forall x \geq 0\}.$

Copositive programming is an elegant and efficient way of stating hard problems.

Checking copositivity is very hard.

Simple Copositivity Criteria

$\text{PSD}_n + \text{NN}_n \subseteq \text{CoP}_n.$

$$\begin{bmatrix} 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \end{bmatrix}$$

Parrilo's Hierarchy

For a symmetric matrix M consider the polynomial

$$p_M(x) = \begin{bmatrix} {x_1}^2 \\ {x_2}^2 \\ \vdots \\ {x_n}^2 \end{bmatrix}^t M \begin{bmatrix} {x_1}^2 \\ {x_2}^2 \\ \vdots \\ {x_n}^2 \end{bmatrix}.$$

Parrilo's Hierarchy

For a symmetric matrix M consider the polynomial

$$p_M(x) = \begin{bmatrix} {x_1}^2 \\ {x_2}^2 \\ \vdots \\ {x_n}^2 \end{bmatrix}^t M \begin{bmatrix} {x_1}^2 \\ {x_2}^2 \\ \vdots \\ {x_n}^2 \end{bmatrix}.$$

M is copositive iff p_M is nonnegative.

Parrilo's Hierarchy

For a symmetric matrix M consider the polynomial

$$p_M(x) = \begin{bmatrix} {x_1}^2 \\ {x_2}^2 \\ \vdots \\ {x_n}^2 \end{bmatrix}^t M \begin{bmatrix} {x_1}^2 \\ {x_2}^2 \\ \vdots \\ {x_n}^2 \end{bmatrix}.$$

M is copositive iff p_M is nonnegative.

Parrilo's hierarchy

$\mathrm{Par}_n^r = \{M \in \mathbb{R}^{n\times n} \ : \ M = M^t, \ \|x\|^{2r} p_M(x) \text{ is sos}\}.$

Parrilo's Hierarchy

For a symmetric matrix M consider the polynomial

$$p_M(x) = \begin{bmatrix} {x_1}^2 \\ {x_2}^2 \\ \vdots \\ {x_n}^2 \end{bmatrix}^t M \begin{bmatrix} {x_1}^2 \\ {x_2}^2 \\ \vdots \\ {x_n}^2 \end{bmatrix}.$$

M is copositive iff p_M is nonnegative.

Parrilo's hierarchy

$\mathrm{Par}_n^r = \{M \in \mathbb{R}^{n\times n} \ : \ M = M^t, \ \|x\|^{2r} p_M(x) \text{ is sos}\}$.

$$\mathrm{Par}_n^1 \subseteq \mathrm{Par}_n^2 \subseteq \cdots \subseteq \mathrm{CoP}_n$$

There is assimptotic convergence, but $\mathrm{Par}_5^r \neq \mathrm{CoP}_5$ for any r.

Hierarchy with free multipliers

Parrilo's hierarchy v2.0

$\text{Par}_n^{*,r} = \{M \in \mathbb{R}^{n\times n} \ : \ M = M^t, \ q(x)p_M(x) \text{ is sos}, q(x) \neq 0 \text{ is sos}\}.$

Hierarchy with free multipliers

Parrilo's hierarchy v2.0

$\mathrm{Par}_n^{*,r} = \{M \in \mathbb{R}^{n\times n} \;:\; M = M^t,\; q(x)p_M(x) \text{ is sos}, q(x) \neq 0 \text{ is sos}\}.$

Checking membership in $\mathrm{Par}_n^{*,r}$ is relatively easy (semidefinite programming).

Hierarchy with free multipliers

Parrilo's hierarchy v2.0

$$\text{Par}_n^{*,r} = \{M \in \mathbb{R}^{n\times n} \;:\; M = M^t,\; q(x)p_M(x) \text{ is sos}, q(x) \neq 0 \text{ is sos}\}.$$

Checking membership in $\text{Par}_n^{*,r}$ is relatively easy (semidefinite programming).

However it is not even clear when is $\text{Par}_n^{*,r}$ convex.

Hierarchy with free multipliers

Parrilo's hierarchy v2.0

$\text{Par}_n^{*,r} = \{M \in \mathbb{R}^{n\times n} \ : \ M = M^t, \ q(x)p_M(x) \text{ is sos}, q(x) \neq 0 \text{ is sos}\}.$

Checking membership in $\text{Par}_n^{*,r}$ is relatively easy (semidefinite programming).

However it is not even clear when is $\text{Par}_n^{*,r}$ convex.

Finite Convergence

For all n there exists r such that $\text{Par}_n^{*,r} = \text{CoP}_n$. In particular $\text{Par}_5^{*,1} = \text{CoP}_5$.

Constrained polynomial optimization

Constrained Problem

$p_{\min} = \min_x p(x) \text{ s.t. } g_i(x) = 0, \; i = 1, \cdots, r.$

Constrained polynomial optimization

Constrained Problem

$p_{\min} = \min_x p(x) \text{ s.t. } g_i(x) = 0, \; i = 1, \cdots, r.$

An equivalent formulation

Constrained Problem

$p_{\min} = \max_\lambda \lambda \text{ s.t. } p(x) - \lambda \geq 0 \text{ for all } x \text{ s.t. } g_i(x) = 0, \; i = 1, \cdots, r.$

Constrained polynomial optimization

Constrained Problem

$p_{\min} = \min_x p(x) \text{ s.t. } g_i(x) = 0, \; i = 1, \cdots, r.$

An equivalent formulation

Constrained Problem

$p_{\min} = \max_\lambda \lambda \text{ s.t. } p(x) - \lambda \geq 0 \text{ for all } x \text{ s.t. } g_i(x) = 0, \; i = 1, \cdots, r.$

We can now apply sums of squares

Constrained Problem Relaxation

$p_{sos} = \max_\lambda \lambda \text{ s.t. } p(x) - \lambda + \sum q_i(x) g_i(x)$ is sos, for some q_i.

Constrained polynomial optimization

Constrained Problem

$p_{\min} = \min_x p(x)$ s.t. $g_i(x) = 0, \ i = 1, \cdots, r.$

An equivalent formulation

Constrained Problem

$p_{\min} = \max_\lambda \lambda$ s.t. $p(x) - \lambda \geq 0$ for all x s.t. $g_i(x) = 0, \ i = 1, \cdots, r.$

We can now apply sums of squares

Constrained Problem Relaxation

$p_{sos} = \max_\lambda \lambda$ s.t. $p(x) - \lambda + \sum q_i(x) g_i(x)$ is sos, for some q_i.

Degree bounds are needed.

Costrained polynomial optimization (continued)

Lasserre Hierarchy

$p_{\text{sos}}^k = \max_\lambda \lambda$ s.t. $p(x) - \lambda + \sum q_i(x)g_i(x)$ is sos, for some polynomials q_i, with degree of p and $q_i g_i$ at most $2k$.

Costrained polynomial optimization (continued)

Lasserre Hierarchy

$p_{\text{sos}}^k = \max_\lambda \lambda$ s.t. $p(x) - \lambda + \sum q_i(x)g_i(x)$ is sos, for some polynomials q_i, with degree of p and q_ig_i at most $2k$.

Again we can adapt this hierarchy to use multipliers

Lasserre Hierarchy v 2.0

$p_{\text{sos}}^{j,k} = \max_\lambda \lambda$ s.t.

$$(1 + q(x))(p(x) - \lambda) + \sum q_i(x)g_i(x)$$

is sos, for some polynomials q_i, with degree of p and q_ig_i at most $2k$ and $q(x)$ a sum of squares of degree at most $2j$.

Costrained polynomial optimization (continued)

Lasserre Hierarchy

$p_{sos}^{k} = \max_\lambda \lambda$ s.t. $p(x) - \lambda + \sum q_i(x) g_i(x)$ is sos, for some polynomials q_i, with degree of p and $q_i g_i$ at most $2k$.

Again we can adapt this hierarchy to use multipliers

Lasserre Hierarchy v 2.0

$p_{sos}^{j,k} = \max_\lambda \lambda$ s.t.

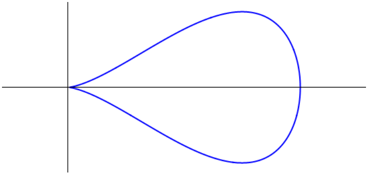
$$(1 + q(x))(p(x) - \lambda) + \sum q_i(x) g_i(x)$$

is sos, for some polynomials q_i, with degree of p and $q_i g_i$ at most $2k$ and $q(x)$ a sum of squares of degree at most $2j$.

Same advantages and disadvantage as before.

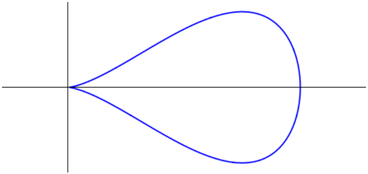
Example

Consider the teardrop curve given by $x^4 - x^3 + y^2 = 0$.



Example

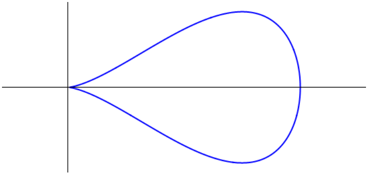
Consider the teardrop curve given by $x^4 - x^3 + y^2 = 0$.



Let $p(x) = x$

Example

Consider the teardrop curve given by $x^4 - x^3 + y^2 = 0$.

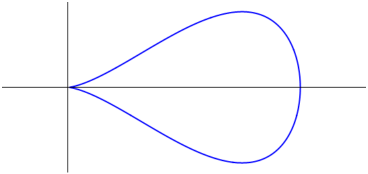


Let $p(x) = x$ then

$$p^2_{\text{sos}} = -0.1250,$$

Example

Consider the teardrop curve given by $x^4 - x^3 + y^2 = 0$.

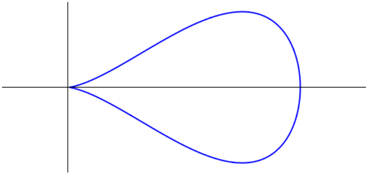


Let $p(x) = x$ then

$$p^2_{\text{sos}} = -0.1250, \quad p^3_{\text{sos}} = -0.0208,$$

Example

Consider the teardrop curve given by $x^4 - x^3 + y^2 = 0$.

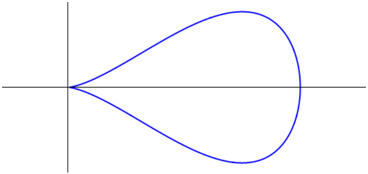


Let $p(x) = x$ then

$$p^2_{\text{sos}} = -0.1250, \quad p^3_{\text{sos}} = -0.0208, \quad p^4_{\text{sos}} = -0.0092, \quad \ldots$$

Example

Consider the teardrop curve given by $x^4 - x^3 + y^2 = 0$.



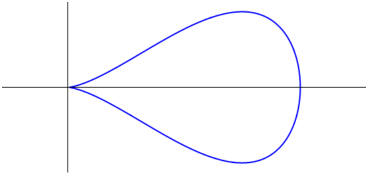
Let $p(x) = x$ then

$$p_{sos}^2 = -0.1250, \quad p_{sos}^3 = -0.0208, \quad p_{sos}^4 = -0.0092, \quad \ldots$$

However $p_{sos}^{1,2} = p_{min} = 0$.

Example

Consider the teardrop curve given by $x^4 - x^3 + y^2 = 0$.



Let $p(x) = x$ then

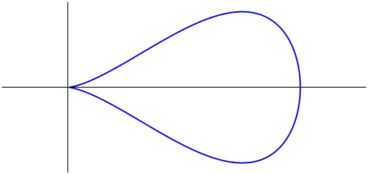
$$p^2_{sos} = -0.1250, \quad p^3_{sos} = -0.0208, \quad p^4_{sos} = -0.0092, \quad \ldots$$

However $p^{1,2}_{sos} = p_{min} = 0$. In fact

$$x^2 \cdot x = x^4 + y^2 \text{ modulo } I.$$

Example

Consider the teardrop curve given by $x^4 - x^3 + y^2 = 0$.



Let $p(x) = x$ then

$$p^2_{\text{sos}} = -0.1250, \quad p^3_{\text{sos}} = -0.0208, \quad p^4_{\text{sos}} = -0.0092, \quad \ldots$$

However $p^{1,2}_{\text{sos}} = p_{\min} = 0$. In fact

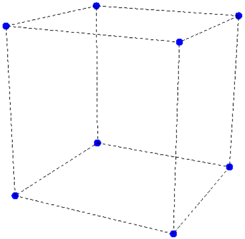
$$x^2 \cdot x = x^4 + y^2 \text{ modulo } I.$$

Multipliers make the relaxations less sensitive to singularities.

The n-cube

We are interested in the n-cube:

$$C_n = \{0,1\}^n = \{x \in \mathbb{R}^n \ : \ x_i^2 - x_i = 0, \ i = 1, \cdots, n\}.$$

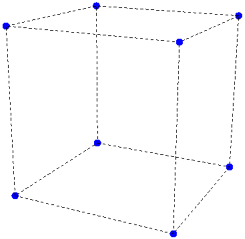


Cube C_3

The n-cube

We are interested in the n-cube:

$$C_n = \{0,1\}^n = \{x \in \mathbb{R}^n \;:\; x_i^2 - x_i = 0,\; i = 1, \cdots, n\}.$$



Cube C_3

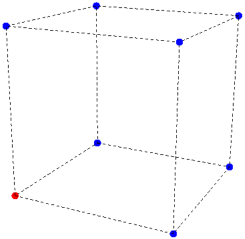
S_n acts on C_n by permuting coordinates, and if p is symmetric, it will be completely characterized by its evaluation at the levels T_k of the cube:

$$T_k = \{x \in C_n \;:\; \sum x_i = k\}.$$

The n-cube

We are interested in the n-cube:

$$C_n = \{0,1\}^n = \{x \in \mathbb{R}^n \ : \ x_i^2 - x_i = 0, \ i = 1, \cdots, n\}.$$



Level T_0

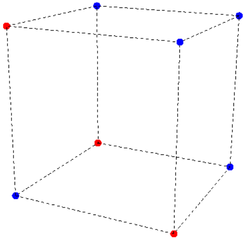
S_n acts on C_n by permuting coordinates, and if p is symmetric, it will be completely characterized by its evaluation at the levels T_k of the cube:

$$T_k = \{x \in C_n \ : \ \sum x_i = k\}.$$

The n-cube

We are interested in the n-cube:

$$C_n = \{0,1\}^n = \{x \in \mathbb{R}^n \ : \ x_i^2 - x_i = 0, \ i = 1, \cdots, n\}.$$



Level T_1

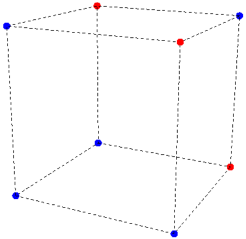
S_n acts on C_n by permuting coordinates, and if p is symmetric, it will be completely characterized by its evaluation at the levels T_k of the cube:

$$T_k = \{x \in C_n \ : \ \sum x_i = k\}.$$

The n-cube

We are interested in the n-cube:

$$C_n = \{0,1\}^n = \{x \in \mathbb{R}^n \; : \; x_i^2 - x_i = 0, \; i = 1, \cdots, n\}.$$



Level T_2

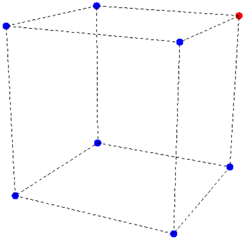
S_n acts on C_n by permuting coordinates, and if p is symmetric, it will be completely characterized by its evaluation at the levels T_k of the cube:

$$T_k = \{x \in C_n \; : \; \sum x_i = k\}.$$

The n-cube

We are interested in the n-cube:

$$C_n = \{0,1\}^n = \{x \in \mathbb{R}^n \ : \ x_i^2 - x_i = 0, \ i = 1, \cdots, n\}.$$



Level T_3

S_n acts on C_n by permuting coordinates, and if p is symmetric, it will be completely characterized by its evaluation at the levels T_k of the cube:

$$T_k = \{x \in C_n \ : \ \sum x_i = k\}.$$

Main Result 1 - Bad news

Let p be a symmetric square-free polynomial attaining its minimum over C_n at level T_k, with $\deg p \leq k \leq n/2$.

Main Result 1 - Bad news

Let p be a symmetric square-free polynomial attaining its minimum over C_n at level T_k, with $\deg p \leq k \leq n/2$.

Theorem

If T_k is not a local extreme of p over $\mathbb{R}^n$ (seen as a polynomial in $\sum x_i$) then $p_{\min} > p_{\text{sos}}^{k-r,k}$, where $r = \lceil (\deg p)/2 \rceil$.

Main Result 1 - Bad news

Let p be a symmetric square-free polynomial attaining its minimum over C_n at level T_k, with $\deg p \leq k \leq n/2$.

Theorem

If T_k is not a local extreme of p over $\mathbb{R}^n$ (seen as a polynomial in $\sum x_i$) then $p_{\min} > p_{\text{sos}}^{k-r,k}$, where $r = \lceil (\deg p)/2 \rceil$.

This means that if the minimizer of p is "simple enough" and is close to the central levels of the cube, we need high level sos relaxations.

Notes on the result

The proof reduces to this lemma.

Lemma

If p has degree d and vanishes at T_k with $d \leq k \leq n - d$ then

$$p = (k - \sum x_i)q \mod I_n,$$

with $\deg q < \deg p$.

Notes on the result

The proof reduces to this lemma.

Lemma

If p has degree d and vanishes at T_k with $d \leq k \leq n - d$ then

$$p = (k - \sum x_i)q \mod I_n,$$

with $\deg q < \deg p$.

This is a divisibility result. Surprisingly, the only proof we know uses representation theory.

Sketch of Proof:

Consider the action of S_n in $\mathbb{R}[I]_k$.

Sketch of Proof:

Consider the action of S_n in $\mathbb{R}[I]_k$. It decomposes:

$$\mathbb{R}[I]_k = \mathbb{R}[I]_{=0} \oplus \mathbb{R}[I]_{=1} \oplus \mathbb{R}[I]_{=2} \oplus \cdots \oplus \mathbb{R}[I]_{=k}$$

Sketch of Proof:

Consider the action of S_n in $\mathbb{R}[I]_k$. It decomposes:

$$
\begin{array}{ccccccccccc}
\mathbb{R}[I]_k = & \mathbb{R}[I]_{=0} & \oplus & \mathbb{R}[I]_{=1} & \oplus & \mathbb{R}[I]_{=2} & \oplus & \cdots & \oplus & \mathbb{R}[I]_{=k} \\
& \wr\Vert & & \wr\Vert & & \wr\Vert & & & & \wr\Vert \\
& H_{n,0} & & H_{n,0} & & H_{n,0} & & \cdots & & H_{n,0} \\
& & & \oplus & & \oplus & & & & \oplus \\
& & & H_{n-1,1} & & H_{n-1,1} & & \cdots & & H_{n-1,1} \\
& & & & & \oplus & & & & \oplus \\
& & & & & H_{n-2,2} & & \cdots & & H_{n-2,2} \\
& & & & & & \ddots & & & \vdots \\
& & & & & & & & & H_{n-k,k}
\end{array}
$$

Sketch of Proof:

Consider the action of S_n in $\mathbb{R}[I]_k$. It decomposes:

$$\begin{array}{cccccccccc}
\mathbb{R}[I]_k = & \mathbb{R}[I]_{=0} & \oplus & \mathbb{R}[I]_{=1} & \oplus & \mathbb{R}[I]_{=2} & \oplus & \cdots & \oplus & \mathbb{R}[I]_{=k} \\
& \wr\| & & \wr\| & & \wr\| & & & & \wr\| \\
& \color{red}{H_{n,0}} & & H_{n,0} & & H_{n,0} & & \cdots & & H_{n,0} \\
& & & \oplus & & \oplus & & & & \oplus \\
& & & \color{red}{H_{n-1,1}} & & H_{n-1,1} & & \cdots & & H_{n-1,1} \\
& & & & & \oplus & & & & \oplus \\
& & & & & \color{red}{H_{n-2,2}} & & \cdots & & H_{n-2,2} \\
& & & & & & \ddots & & & \vdots \\
& & & & & & & & & \color{red}{H_{n-k,k}}
\end{array}$$

Let M_j be the first copy of $H_{n-j,j}$ to appear,

Sketch of Proof:

Consider the action of S_n in $\mathbb{R}[I]_k$. It decomposes:

$$\begin{array}{ccccccccccc}
\mathbb{R}[I]_k = & \mathbb{R}[I]_{=0} & \oplus & \mathbb{R}[I]_{=1} & \oplus & \mathbb{R}[I]_{=2} & \oplus & \cdots & \oplus & \mathbb{R}[I]_{=k} \\
 & \wr\| & & \wr\| & & \wr\| & & & & \wr\| \\
 & H_{n,0} & & H_{n,0} & & H_{n,0} & & \cdots & & H_{n,0} \\
 & & & \oplus & & \oplus & & & & \oplus \\
 & & & H_{n-1,1} & & H_{n-1,1} & & \cdots & & H_{n-1,1} \\
 & & & & & \oplus & & & & \oplus \\
 & & & & & H_{n-2,2} & & \cdots & & H_{n-2,2} \\
 & & & & & & \ddots & & & \vdots \\
 & & & & & & & & & H_{n-k,k}
\end{array}$$

Let M_j be the first copy of $H_{n-j,j}$ to appear, then

$$\mathbb{R}[I]_k = \bigoplus_{j=0}^{k} M_j \oplus (k - \sum x_i) M_j \oplus \cdots \oplus (k - \sum x_i)^{k-j} M_j$$

Sketch of Proof:

Consider the action of S_n in $\mathbb{R}[I]_k$. It decomposes:

$$\begin{array}{ccccccccccc}
\mathbb{R}[I]_k = & \mathbb{R}[I]_{=0} & \oplus & \mathbb{R}[I]_{=1} & \oplus & \mathbb{R}[I]_{=2} & \oplus & \cdots & \oplus & \mathbb{R}[I]_{=k} \\
 & \wr\| & & \wr\| & & \wr\| & & & & \wr\| \\
 & H_{n,0} & & H_{n,0} & & H_{n,0} & & \cdots & & H_{n,0} \\
 & & & \oplus & & \oplus & & & & \oplus \\
 & & & H_{n-1,1} & & H_{n-1,1} & & \cdots & & H_{n-1,1} \\
 & & & & & \oplus & & & & \oplus \\
 & & & & & H_{n-2,2} & & \cdots & & H_{n-2,2} \\
 & & & & & & \ddots & & & \vdots \\
 & & & & & & & & & H_{n-k,k}
\end{array}$$

Let M_j be the first copy of $H_{n-j,j}$ to appear, then

$$\mathbb{R}[I]_k = \bigoplus_{j=0}^{k} M_j \oplus (k - \sum x_i) M_j \oplus \cdots \oplus (k - \sum x_i)^{k-j} M_j$$

and is enough to check that M_j does not vanish at T_k.

Application 1 - MaxCut

Recall that the maxcut problem over K_n can be reduced to

Binary polynomial formulation of MaxCut

$$\max p(x) = \sum_{i \neq j} (1 - x_i) x_j \text{ s.t. } x \in C_n$$

Application 1 - MaxCut

Recall that the maxcut problem over K_n can be reduced to

Binary polynomial formulation of MaxCut

$$\max p(x) = \sum_{i \neq j} (1 - x_i) x_j \text{ s.t. } x \in C_n$$

Laurent has proved that Lasserre relaxations are of limited use.

Laurent

For $n = 2k + 1$, $p_{sos}^k > p_{max}$.

Application 1 - MaxCut

Recall that the maxcut problem over K_n can be reduced to

Binary polynomial formulation of MaxCut

$$\max p(x) = \sum_{i \neq j} (1 - x_i) x_j \text{ s.t. } x \in C_n$$

Laurent has proved that Lasserre relaxations are of limited use.

Laurent

For $n = 2k + 1$, $p_{\text{sos}}^k > p_{\text{max}}$.

Note that p attains its maximum in C_n at T_k and T_{k+1}, which are not local maxima of p over $\mathbb{R}^n$.

First corollary of main result 1

For $n = 2k + 1$, ${p_{\text{sos}}}^{k-1,k} > p_{\text{max}}$.

Application 2 - Global Optimization

Let p be any polynomial in $\mathbb{R}^n$.

Application 2 - Global Optimization

Let p be any polynomial in $\mathbb{R}^n$.

Artin (Hilbert's 17th Problem)

For some l, k, $p_{\text{sos}}^{l,k} = p_{\min}$.

Application 2 - Global Optimization

Let p be any polynomial in $\mathbb{R}^n$.

Artin (Hilbert's 17th Problem)

For some l, k, $p_{\text{sos}}^{l,k} = p_{\min}$.

We also expect that these l, k should be very high. However there were no examples for such behavior.

Application 2 - Global Optimization

Let p be any polynomial in $\mathbb{R}^n$.

Artin (Hilbert's 17th Problem)

For some l, k, $p_{\text{sos}}^{l,k} = p_{\min}$.

We also expect that these l, k should be very high. However there were no examples for such behavior.

Second corollary of main result 1

For any k there is a degree 4 polynomial in $\mathbb{R}^{2k+1}$ for which $p_{\min} \neq p_{\text{sos}}^{k-2,k}$.

Application 2 - Global Optimization

Let p be any polynomial in $\mathbb{R}^n$.

Artin (Hilbert's 17th Problem)

For some l, k, $p_{\text{sos}}^{l,k} = p_{\min}$.

We also expect that these l, k should be very high. However there were no examples for such behavior.

Second corollary of main result 1

For any k there is a degree 4 polynomial in $\mathbb{R}^{2k+1}$ for which $p_{\min} \neq p_{\text{sos}}^{k-2,k}$.

This is proven by a perturbed extension of the polynomial on the previous example.

$$p = \sum_{i \neq j} (1 - x_i) x_j$$

Application 2 - Global Optimization

Let p be any polynomial in $\mathbb{R}^n$.

Artin (Hilbert's 17th Problem)

For some l, k, $p_{\text{sos}}^{l,k} = p_{\min}$.

We also expect that these l, k should be very high. However there were no examples for such behavior.

Second corollary of main result 1

For any k there is a degree 4 polynomial in $\mathbb{R}^{2k+1}$ for which $p_{\min} \neq p_{\text{sos}}^{k-2,k}$.

This is proven by a perturbed extension of the polynomial on the previous example.

$$p = \sum_{i \neq j} (1 - x_i) x_j + \varepsilon$$

Application 2 - Global Optimization

Let p be any polynomial in $\mathbb{R}^n$.

Artin (Hilbert's 17th Problem)

For some l, k, $p_{\text{sos}}^{l,k} = p_{\min}$.

We also expect that these l, k should be very high. However there were no examples for such behavior.

Second corollary of main result 1

For any k there is a degree 4 polynomial in $\mathbb{R}^{2k+1}$ for which $p_{\min} \neq p_{\text{sos}}^{k-2,k}$.

This is proven by a perturbed extension of the polynomial on the previous example.

$$p = \sum_{i \neq j} (1 - x_i) x_j + \varepsilon + A \sum_i ({x_i}^2 - x_i)^2$$

Main Result 2 - Not so bad news

We have showed lower bounds to the effectiveness of sos for binary polynomial programming. Luckily we also can show some upper bounds.

Main Result 2 - Not so bad news

We have showed lower bounds to the effectiveness of sos for binary polynomial programming. Luckily we also can show some upper bounds.

Theorem

Let p be a non constant quadratic polynomial in $\mathbb{R}^{2k+1}$, then $p_{\text{min}} = p_{\text{sos}}^{k+1,k+2}$ (over the cube).

Main Result 2 - Not so bad news

We have showed lower bounds to the effectiveness of sos for binary polynomial programming. Luckily we also can show some upper bounds.

Theorem

Let p be a non constant quadratic polynomial in $\mathbb{R}^{2k+1}$, then $p_{\min} = p_{\text{sos}}^{k+1,k+2}$ (over the cube).

The proof is based in dimension counting.

Application - MaxCut revisited

Consider the weighted maxcut formulation.

Binary polynomial formulation of MaxCut

$$\max p_\omega(x) = \sum_{i \neq j} \omega_{ij}(1 - x_i)x_j \text{ s.t. } x \in C_n,$$

where ω_{ij} is the weight of edge $\{i, j\}$.

Application - MaxCut revisited

Consider the weighted maxcut formulation.

Binary polynomial formulation of MaxCut

$$\max p_\omega(x) = \sum_{i \neq j} \omega_{ij}(1 - x_i)x_j \text{ s.t. } x \in C_n,$$

where ω_{ij} is the weight of edge $\{i, j\}$.

The negative result proved by Laurent has an opposed positive conjecture.

Conjecture (Laurent)

If $n = 2k + 1$, $(p_\omega)_{\min} = (p_\omega)^{k+1}_{\text{sos}}$ for all weights.

Application - MaxCut revisited

Consider the weighted maxcut formulation.

Binary polynomial formulation of MaxCut

$$\max p_\omega(x) = \sum_{i \neq j} \omega_{ij}(1 - x_i)x_j \text{ s.t. } x \in C_n,$$

where ω_{ij} is the weight of edge $\{i, j\}$.

The negative result proved by Laurent has an opposed positive conjecture.

Conjecture (Laurent)

If $n = 2k + 1$, $(p_\omega)_{\min} = (p_\omega)^{k+1}_{\text{sos}}$ for all weights.

A weaker version can now be proved.

Corollary of main result 2

If $n = 2k + 1$, $(p_\omega)_{\min} = (p_\omega)^{k+1,k+2}_{\text{sos}}$ for all weights.

Open Questions

- Show that for every r there exists n such that $\mathrm{Par}_n^{*,r} \neq \mathrm{CoP}_n$. (Adapt the polynomial we have?)

Open Questions

- Show that for every r there exists n such that $\mathrm{Par}_n^{*,r} \neq \mathrm{CoP}_n$. (Adapt the polynomial we have?)
- Convexity of $\mathrm{Par}_n^{*,r}$.

Open Questions

- Show that for every r there exists n such that $\mathrm{Par}_n^{*,r} \neq \mathrm{CoP}_n$. (Adapt the polynomial we have?)
- Convexity of $\mathrm{Par}_n^{*,r}$.
- How to use $\mathrm{Par}_n^{*,r}$ in general copositive programming.

Open Questions

- Show that for every r there exists n such that $\mathrm{Par}_n^{*,r} \neq \mathrm{CoP}_n$. (Adapt the polynomial we have?)
- Convexity of $\mathrm{Par}_n^{*,r}$.
- How to use $\mathrm{Par}_n^{*,r}$ in general copositive programming.
- Any progress on sos/sdp hardness of matching.

The End

Thank You