Lecture 6

Modified Sum-of-Squares Relaxations for Large Scale Optimizations

Fall 2019

Ashkan Jasour
jasour@mit.edu
jasour.mit.edu
Moment-SOS Relaxations

Nonlinear-Nonconvex Optimization ➔ Convexification

- SOS Relaxation
- Moment Relaxation

Dual optimization

Semidefinite Program
Moment-SOS Relaxations: Applications in Robotics and Control

Motion Planning

- A. Majumdar, R. Tedrake, “Funnel libraries for real-time robust feedback motion planning”, international journal of robotics and research(IJRR), Volume: 36 issue: 8, page(s): 947-982, 2017

Planning and Controllers for UAV


Legged Robots


Real-Time Planning


Controller Design

Moment-SOS Relaxations: Applications in Robotics and Control

Validation


• S. Shen, R. Tedrake, “Compositional Verification of Large-Scale Nonlinear Systems via Sums-of-Squares Optimization”, American Control Conference (ACC) 2018

Environment Representation

• A. A. Ahmadi, G. Hall, A. Makadia, and V. Sindhwani, “Sum of Squares Polynomials and Geometry of 3D Environments” Robotics: Science and Systems, 2017

Control and Analysis

• M. Korda, D. Henrion, C. N. Jones. Controller design and region of attraction estimation for nonlinear dynamical systems. , October 2013, updated in March 2014,


- What is the cost of convexification?
**Nonlinear Optimization:** variables \((x_1, x_2)\)

\[
P^* = \min_{x \in \mathbb{R}^2} \frac{1}{3} x_1^6 - \frac{21}{10} x_1^4 + 4 x_1^2 + x_1 x_2 + 4 x_2^4 - 4 x_2^2 + \frac{3}{2}
\]

subject to \(x \in K = \{x \in \mathbb{R}^2 : -\frac{1}{16} x_1^4 + \frac{1}{4} x_1^3 - \frac{1}{4} x_1^2 - \frac{9}{100} x_2^2 + \frac{29}{400} \geq 0\}\)
Nonlinear Optimization: variables \((x_1, x_2)\)

\[
P^* \in \mathbb{R}^2 \quad \text{minimize} \quad \frac{1}{3} x_1^6 - \frac{21}{10} x_1^4 + 4 x_1^2 + x_1 x_2 + 4 x_2^4 - 4 x_2^2 + \frac{3}{2}
\]

subject to \(x \in \mathbf{K} = \{x \in \mathbb{R}^2 : -\frac{1}{16} x_1^4 + \frac{1}{4} x_1^3 - \frac{1}{4} x_1^2 - \frac{9}{100} x_2^2 + \frac{29}{400} \geq 0\}\)

Moment SDP: variables are moments \(y_{\alpha_1 \alpha_2} = \mathbb{E}[x_1^{\alpha_1} x_2^{\alpha_2}]\) \(y = [y_{\alpha}, \alpha = 0, \ldots, 6]\)

\[
P^*_{\text{mom}} = \min_y \quad \frac{1}{3} y_{60} - \frac{21}{10} y_{40} + 4 y_{20} + y_{11} + 4 y_{04} - 4 y_{02} + \frac{3}{2} y_{00}
\]

subject to \(y_{00} = 1\)

\(\mathcal{M}_3(y) \succeq 0, \mathcal{M}_8(gg) \succeq 0\)

- Number of Moments in \(\mathbb{R}^n\) up to order \(2d\):

\[
\binom{n + 2d}{2d} = \frac{(2 + 6)!}{2!6!} = 28
\]
**Nonlinear Optimization:** variables \((x_1, x_2)\)

\[
P^* = \text{minimize}_{x \in \mathbb{R}^2} \quad \frac{1}{3}x_1^6 - \frac{21}{10}x_1^4 + 4x_1^2 + x_1x_2 + 4x_2^4 - 4x_2^2 + \frac{3}{2}
\]

subject to \(x \in K = \{x \in \mathbb{R}^2 : -\frac{1}{16}x_1^4 + \frac{1}{4}x_1^3 - \frac{1}{4}x_1^2 - \frac{9}{100}x_2^2 + \frac{29}{400} \geq 0\}\)

**Moment SDP:** variables are moments \(y_{\alpha_1\alpha_2} = E[x_1^{\alpha_1}x_2^{\alpha_2}]\) \(y = [y_\alpha, \alpha = 0, ..., 6]\)

\[
P^{*\text{mom}} = \text{minimize}_y \quad \frac{1}{3}y_6 - \frac{21}{10}y_4 + 4y_2 + y_1 + 4y_{04} - 4y_{02} + \frac{3}{2}y_0
\]

subject to \(y_0 = 1\)
\[M_3(y) \succeq 0, \quad M_{3-2}(gg) \succeq 0\]

- Number of Moments in \(\mathbb{R}^n\) up to order 2\(d\):
  \[
  \binom{n+2d}{n} = \frac{(2+6)!}{2!6!} = 28
  \]

**SOS SDP:** variables are coefficients of polynomial

\[
P^{*\text{sos}} = \text{maximize}_{\gamma \in \mathbb{R}, \sigma_1} \quad \gamma
\]

subject to \(p(x) - \gamma - \sigma_1(x)g(x) \in SOS_6\)
\[\sigma_1(x) \in SOS_2\]

- Number of coefficients of a 2\(d\)-degree polynomial in \(\mathbb{R}^n\):
  \[
  \binom{n+2d}{n} = \frac{(2+6)!}{2!6!} = 28
  \]
What is the cost of convexification?

Convexification increases the dimension of the search space.

- Number of variables of the original nonlinear optimization: $n$
- Number of variables Moment SDP: $\binom{n+2d}{n}$
Moment-SOS Relaxations

Convexification increases the dimension of the search space.

- Number of variables of the original nonlinear optimization: $n$
- Number of variables Moment SDP: $\binom{n+2d}{n}$

Pros:
- Moment-SOS relaxations solve difficult and challenging mathematical problems.
- They provide insights into challenging problems where no other solid and comprehensive approach exist.
  (e.g., existing approaches for nonlinear robust and chance constrained optimizations work for particular class of problems,...).
Moment-SOS Relaxations

- Current SDP solvers are interior-point based solvers.
- In the absence of problem structure, sum of squares problems are currently limited, roughly speaking, to a several thousands variables (variables in SDP).

- How to address large scale problems?
Moment-SOS Relaxations

How to address large scale problems?

1) Modified SOS optimization to generate i) smaller SDP’s or ii) other types of convex constraints like LP.

   Approaches:
   i) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS),
   ii) Bounded degree SOS (BSOS)
Moment-SOS Relaxations

How to address large scale problems?

1) Modified SOS optimization to generate i) smaller SDP’s or ii) other types of convex constraints like LP.

   Approaches:
   i) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS),
   ii) Bounded degree SOS (BSOS)

2) Take advantage of structure of the problem (sparsity) to generate smaller SDP’s.

   Approaches:
   i) Spars Sum-of-Squares Optimization (SSOS)
   ii) Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)
Moment-SOS Relaxations

How to address large scale problems?

1) Modified SOS optimization to generate i) smaller SDP’s or ii) other types of convex constraints like LP.
   Approaches:
   i) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS),
   ii) Bounded degree SOS (BSOS)

2) Take advantage of structure of the problem (sparsity) to generate smaller SDP’s.
   Approaches:
   i) Spars Sum-of-Squares Optimization (SSOS)
   ii) Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)

3) Efficient Algorithms for Large Scale SDP’s (Lecture 9)
Moment-SOS Relaxations

How to address large scale problems?

1) Modified SOS optimization to generate i) smaller SDP’s or ii) other types of convex constraints like LP.
   Approaches:
   i) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS),
   ii) Bounded degree SOS (BSOS)

2) Take advantage of structure of the problem (sparsity) to generate smaller SDP’s.
   Approaches:
   i) Spars Sum-of-Squares Optimization (SSOS)
   ii) Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)

3) Efficient Algorithms for Large Scale SDP’s (Lecture 9)

4) Reformulate original optimization problem to reduce the size of the optimization (Lectures 10 and 11)
Topics

1) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS)
Modified SOS optimization that results in LP and Second order cone program

Applications:
Control and analyze of high dimensional systems

Topics

1) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS)
Modified SOS optimization that results in LP and Second order cone program

2) Bounded degree SOS (BSOS)
Modified SOS optimization that results in smaller SDP’s.

• Jean B. Lasserre, Kim-Chuan Toh, Shouguang Yang, “A bounded degree SOS hierarchy for polynomial optimization”, EURO Journal on Computational Optimization March 2017, Volume 5, Issue 1–2, pp 87–117
Topics

1) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS)
   Modified SOS optimization that results in LP and Second order cone program

2) Bounded degree SOS (BSOS)
   Modified SOS optimization that results in smaller SDP’s.

3) Sparse Sum-of-Squares Optimization (SSOS)
   Takes advantage of sparsity of the original problem to generate smaller SDP.


Topics

1) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS)
   Modified SOS optimization that results in LP and Second order cone program

2) Bounded degree SOS (BSOS)
   Modified SOS optimization that results in smaller SDP’s.

3) Sparse Sum-of-Squares Optimization (SSOS)
   Takes advantage of sparsity of the original problem to generate smaller SDP.

4) Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)
   Combination of 2 and 3

Topics

1) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS)
Modified SOS optimization that results in LP and Second order cone program

2) Bounded degree SOS (BSOS)
Modified SOS optimization that results in smaller SDP’s.

3) Sparse Sum-of-Squares Optimization (SSOS)
Takes advantage of sparsity of the original problem to generate smaller SDP.

4) Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)
Combination of 2 and 3
(Scaled) Diagonally-Dominant SOS Optimization (DSOS, SDSOS)

Nonlinear Optimization and Nonnegative polynomials

**Unconstrained Optimization:**

\[
\begin{align*}
&\text{minimize} \quad p(x) \\
&\text{subject to} \quad p(x) \in \mathbb{R}[x]
\end{align*}
\]

\[
\begin{align*}
\text{maximize} \quad \gamma & \\
\text{subject to} \quad p(x) - \gamma \geq 0, \quad \forall x \in \mathbb{R}^n
\end{align*}
\]

Convex optimization

Polynomial Nonnegativity Constraint

Replace with convex constraints

**Constrained Optimization:**

\[
\begin{align*}
&\text{minimize} \quad p(x) \\
&\text{subject to} \quad g_i(x) \geq 0, \quad i = 1, \ldots, m
\end{align*}
\]

\[
\begin{align*}
&\text{maximize} \quad \gamma & \\
&\text{subject to} \quad p(x) - \gamma \geq 0, \quad \forall x \in \{x \in \mathbb{R}^n : g_i(x) \geq 0, \quad i = 1, \ldots, m\}
\end{align*}
\]

Convex optimization

Polynomial Nonnegativity Constraint

Replace with convex constraints
Sum of squares Polynomials

Polynomial $p(x)$ is **sum of squares (SOS)** polynomial if:

- it can be written as a finite sum of squares of other polynomials.

$$ p(x) \in \mathbb{R}[x] \quad \text{SOS} \quad p(x) = \sum_{i=1}^{\ell} h_i^2(x) \quad h_i(x) \in \mathbb{R}[x], \quad i = 1, \ldots, \ell $$

- If polynomial $p(x)$ is **SOS**, then it is $p(x) \geq 0$ for all

**PSD Matrix representation of SOS polynomials**

$$ p(x) = B(x)^T Q B(x) \quad Q \in S^n, \quad Q \succeq 0 \quad \text{where} \quad B(x): \text{vector of monomials in } x $$
Sum of squares Polynomials

\[ p(x) \in SOS \iff Q \in S^n, \quad Q \succeq 0 \iff \text{Nonnegative Eigenvalues} \iff \text{SDP} \]

PSD Matrix
To avoid SDP and obtain computationally cheap convex optimizations, we obtain relaxed condition for PSD matrices.

For this, we use the following Results:

1) Gershgorin Circle Theorem
2) Diagonally Dominant Matrix (dd)
Gershgorin Circle Theorem

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3} \]

\[
\begin{align*}
\text{Disk}_1(Q_{11}, R_1) &= |Q_{12}| + |Q_{13}| \\
\text{Disk}_2(Q_{22}, R_2) &= |Q_{21}| + |Q_{23}| \\
\text{Disk}_3(Q_{33}, R_3) &= |Q_{31}| + |Q_{32}| 
\end{align*}
\]

Gershgorin Discs
Gershgorin Circle Theorem

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3} \]

\[
\begin{align*}
Disk_1(Q_{11}, R_1) &= |Q_{12}| + |Q_{13}| \\
Disk_2(Q_{22}, R_2) &= |Q_{21}| + |Q_{23}| \\
Disk_3(Q_{33}, R_3) &= |Q_{31}| + |Q_{32}| 
\end{align*}
\]

- Eigenvalue of \( Q \) lies within the Gershgorin discs.

\[
Q \in \mathbb{R}^{n \times n} \quad \rightarrow \quad \text{Eigenvalues} \in \bigcup_{i=1}^{n} Disk_i(Q_{ii}, R_i = \sum_{i \neq j} |Q_{ij}|), \quad i = 1, \ldots, n
\]
Gershgorin Circle Theorem

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \in \mathbb{R}^{3\times3} \]

- \( Disk_1(Q_{11}, R_1 = |Q_{12}| + |Q_{13}|) \)
- \( Disk_2(Q_{22}, R_2 = |Q_{21}| + |Q_{23}|) \)
- \( Disk_3(Q_{33}, R_3 = |Q_{31}| + |Q_{32}|) \)

- Eigenvalue of \( Q \) lies within the Gershgorin discs.

\[ Q \in \mathbb{R}^{n\times n} \quad \Rightarrow \quad \text{Eigenvalues} \in \bigcup_{i=1}^{n} Disk_i(Q_{ii}, R_i = \sum_{i \neq j} |Q_{ij}|), \quad i = 1, \ldots, n \]

- We use Gershgorin Circle Theorem to obtain relaxed PSD condition in terms of entries of matrices.
Gershgorin Circle Theorem

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3} \]

- Eigenvalue of \( Q \) lies within the Gershgorin discs.

\[
\begin{align*}
    & \text{Disk}_1(Q_{11}, R_1) = |Q_{12}| + |Q_{13}| \\
    & \text{Disk}_2(Q_{22}, R_2) = |Q_{21}| + |Q_{23}| \\
    & \text{Disk}_3(Q_{33}, R_3) = |Q_{31}| + |Q_{32}| \\
\end{align*}
\]

\[
Q \in \mathbb{R}^{n \times n} \quad \Rightarrow \quad \text{Eigenvalues} \in \bigcup_{i=1}^{n} \text{Disk}_i(Q_{ii}, R_i = \sum_{i \neq j} |Q_{ij}|), \quad i = 1, ..., n
\]

- We use Gershgorin Circle Theorem to obtain relaxed PSD condition in terms of entries of matrices.

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{12} & Q_{22} & Q_{23} \\ Q_{13} & Q_{23} & Q_{33} \end{bmatrix} \in S^3 \]

\((Q_{33} - R_3) \quad Q_{33} \quad Q_{11} \quad 0 \quad Q_{22}\)
Eigenvalue of $Q$ lies within the Gershgorin discs.

\[ Q \in \mathbb{R}^{n \times n} \quad \Rightarrow \quad \text{Eigenvalues} \in \bigcup_{i=1}^{n} \text{Disk}_i(Q_{ii}, R_i = \sum_{i \neq j} |Q_{ij}|), \quad i = 1, \ldots, n \]

- We use Gershgorin Circle Theorem to obtain relaxed PSD condition in terms of entries of matrices.

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \in S^3 \quad (Q_{33} - R_3) \quad Q_{33} \quad Q_{11} \quad 0 \quad Q_{22} \]

Smallest Eigenvalue $\geq \min_{i=1,2,3} (Q_{ii} - R_i)$
Gershgorin Circle Theorem

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3} \]

\[ \text{Disk}_1(Q_{11}, R_1 = |Q_{12}| + |Q_{13}|) \]
\[ \text{Disk}_2(Q_{22}, R_2 = |Q_{21}| + |Q_{23}|) \]
\[ \text{Disk}_3(Q_{33}, R_3 = |Q_{31}| + |Q_{32}|) \]

- Eigenvalue of \( Q \) lies within the Gershgorin discs.

\[ Q \in \mathbb{R}^{n \times n} \quad \implies \quad \text{Eigenvalues} \in \bigcup_{i=1}^{n} \text{Disk}_i(Q_{ii}, R_i = \sum_{i \neq j} |Q_{ij}|), \quad i = 1, \ldots, n \]

- We use Gershgorin Circle Theorem to obtain relaxed PSD condition in terms of entries of matrices.

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{12} & Q_{22} & Q_{23} \\ Q_{13} & Q_{23} & Q_{33} \end{bmatrix} \in S^3 \]

\[ (Q_{33} - R_3) \]

PSD \( \implies \) Smallest Eigenvalue \( \geq \min_{i=1,2,3} (Q_{ii} - R_i) \geq 0 \)
Gershgorin Circle Theorem

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3} \]

\[ \text{Disk}_1(Q_{11}, R_1 = |Q_{12}| + |Q_{13}|) \]
\[ \text{Disk}_2(Q_{22}, R_2 = |Q_{21}| + |Q_{23}|) \]
\[ \text{Disk}_3(Q_{33}, R_3 = |Q_{31}| + |Q_{32}|) \]

- Eigenvector of \( Q \) lies within the Gershgorin discs.

\[ Q \in \mathbb{R}^{n \times n} \quad \Rightarrow \quad \text{Eigenvalues} \in \bigcup_{i=1}^{n} \text{Disk}_i(Q_{ii}, R_i = \sum_{i \neq j} |Q_{ij}|), \quad i = 1, \ldots, n \]

- We use Gershgorin Circle Theorem to obtain relaxed PSD condition in terms of entries of matrices.

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{12} & Q_{22} & Q_{23} \\ Q_{13} & Q_{23} & Q_{33} \end{bmatrix} \in S^3 \]

\[ (Q_{33} - R_3) \]

PSD \( \Rightarrow \) Smallest Eigenvalue \( \geq \min_{i=1,2,3} (Q_{ii} - R_i) \geq 0 \)

\[ Q_{11} \geq R_1 = |Q_{12}| + |Q_{13}| \]
\[ Q_{22} \geq R_2 = |Q_{12}| + |Q_{23}| \]
\[ Q_{33} \geq R_3 = |Q_{13}| + |Q_{23}| \]
Gershgorin Circle Theorem

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3} \]

\[ \text{Disk}_1(Q_{11}, R_1 = |Q_{12}| + |Q_{13}|) \]
\[ \text{Disk}_2(Q_{22}, R_2 = |Q_{21}| + |Q_{23}|) \]
\[ \text{Disk}_3(Q_{33}, R_3 = |Q_{31}| + |Q_{32}|) \]

- Eigenvalue of \( Q \) lies within the Gershgorin discs.

\[ Q \in \mathbb{R}^{n \times n} \quad \Rightarrow \quad \text{Eigenvalues} \in \bigcup_{i=1}^{n} \text{Disk}_i(Q_{ii}, R_i = \sum_{i \neq j} |Q_{ij}|), \quad i = 1, \ldots, n \]

- We use Gershgorin Circle Theorem to obtain relaxed PSD condition in terms of entries of matrices.

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \in \mathbb{S}^3 \]

\[ (Q_{33} - R_3) \]

Diagonally Dominant Matrix (dd):

\[ Q \in \mathbb{S}^n \quad Q_{ii} \geq \sum_{i \neq j} |Q_{ij}|, \quad i = 1, \ldots, n \]

\[ Q \in \mathbb{S}^n_{dd} \subset \mathbb{S}^n_+ \]
Nonnegative Polynomials

\[ p(x) \geq 0 \]

\[ p(x) = B^T(x)QB(x) \]

\[ p(x) \in SOS \quad Q \in \mathcal{S}^n_+ \quad \text{Nonnegative Eigenvalues} \quad \text{SDP} \]

Nonnegative Polynomials

\[ p(x) \geq 0 \]

\[ p(x) = B^T(x)QB(x) \]

\[ p(x) \in \text{SOS} \quad \rightarrow \quad Q \in \mathcal{S}_{++}^n \quad \rightarrow \quad \text{Nonnegative Eigenvalues} \quad \rightarrow \quad \text{SDP} \]

PSD Matrix

\[ Q \in \mathcal{S}_{dd}^n \quad \rightarrow \quad Q_{ii} \geq \sum_{i \neq j} |Q_{ij}|, \quad i = 1, \ldots, n \]

Diagonally Dominant Matrix

Nonnegative Polynomials

\[ p(x) \geq 0 \]

\[ p(x) = B^T(x)QB(x) \]

\[ p(x) \in SOS \quad \rightarrow \quad Q \in S^n_+ \quad \rightarrow \quad \text{Nonnegative Eigenvalues} \quad \rightarrow \quad \text{SDP} \]

\[ Q \in S^n_{dd} \quad \rightarrow \quad Q_{ii} \geq \sum_{i \neq j} |Q_{ij}| \frac{z_{ij}}{z_{ij}} \quad i = 1, \ldots, n \]

\[ \begin{align*}
Q_{ii} &\geq \sum_{i \neq j} z_{ij}, \quad i = 1, \ldots, n \\
-z_{ij} &\leq Q_{ij} \leq z_{ij}, \quad \forall i, j \quad i \neq j
\end{align*} \]

Nonnegative Polynomials

\[ p(x) \geq 0 \]

\[ p(x) = B^T(x)QB(x) \]

\[ p(x) \in SOS \quad \text{PSD Matrix} \quad Q \in S^n_+ \quad \text{Nonnegative Eigenvalues} \quad \text{SDP} \]

\[ \begin{align*}
p(x) \in DSOS & \quad \text{Diagonally Dominant Matrix} \\
Q & \in S^n_{dd} \\
Q_{ii} & \geq \sum_{i \neq j}^n |Q_{ij}|, \quad i = 1, \ldots, n \\
- z_{ij} & \leq Q_{ij} \leq z_{ij}, \quad \forall i, j \quad i \neq j
\end{align*} \]

Nonnegative Polynomials

\[ p(x) \geq 0 \]

\[ p(x) = B^T(x)QB(x) \]

\[ p(x) \in SOS \quad Q \in S^+_n \quad \text{Nonnegative Eigenvalues} \quad \longrightarrow \text{SDP} \]

\[ p(x) \in DSOS \quad Q \in S_{dd}^n \quad \text{Diagonally Dominant Matrix} \]

\[ Q_{ii} \geq \sum_{i \neq j} |Q_{ij}|, \quad i = 1, \ldots, n \]

\[ Q_{ii} \geq \sum_{i \neq j} z_{ij}, \quad i = 1, \ldots, n \]

\[ -z_{ij} \leq Q_{ij} \leq z_{ij}, \quad \forall i, j \; i \neq j \]


38

Fall 2019
Unconstrained optimization

$$\min_{x \in \mathbb{R}^n} \quad p(x)$$

maximize $$\gamma$$
subject to $$p(x) - \gamma \geq 0, \quad \forall x \in \mathbb{R}^n$$

**SOS Programming: SOS SDP**

maximize $$\gamma$$
subject to $$p(x) - \gamma = B^T(x)QB(x)$$
$$Q \in S^n_+$$

**DSOS Programming: Linear Program**

maximize $$\gamma$$
subject to $$p(x) - \gamma = B^T(x)QB(x)$$
$$Q \in S^n_{dd}$$
Unconstrained optimization

\[
\begin{align*}
\text{minimize} \quad & p(x) \\
\text{subject to} \quad & p(x) - \gamma \geq 0, \quad \forall x \in \mathbb{R}^n
\end{align*}
\]

SOS Programming: SOS SDP

\[
\begin{align*}
\text{maximize} \quad & \gamma \\
\text{subject to} \quad & p(x) - \gamma = B^T(x)QB(x) \\
& Q \in S^n_+
\end{align*}
\]

DSOS Programming: Linear Program

\[
\begin{align*}
\text{maximize} \quad & \gamma \\
\text{subject to} \quad & p(x) - \gamma = B^T(x)QB(x) \\
& Q \in S^n_{dd}
\end{align*}
\]

Constrained optimization

\[
\begin{align*}
\text{maximize} \quad & p(x) \\
\text{subject to} \quad & x \in K = \{x \in \mathbb{R}^n : g_i(x) \geq 0, \ i = 1, \ldots, m\}
\end{align*}
\]

SOS Programming: SOS SDP

\[
\begin{align*}
\text{maximize} \quad & \gamma \\
\text{subject to} \quad & p(x) - \gamma - \sum_{i=0}^{m} \sigma_i(x)g_i(x) = B_d(x)Q_0B_d(x) \\
& \sigma_i(x) = B^T_{d_i}(x)Q_iB_{d_i}(x), \ i = 1, \ldots, m \\
& Q_i \in S^n_{+}, i = 0, \ldots, m
\end{align*}
\]

DSOS Programming: Linear Program

\[
\begin{align*}
\text{maximize} \quad & \gamma \\
\text{subject to} \quad & p(x) - \gamma - \sum_{i=0}^{m} \sigma_i(x)g_i(x) = B_d(x)Q_0B_d(x) \\
& \sigma_i(x) = B^T_{d_i}(x)Q_iB_{d_i}(x), \ i = 1, \ldots, m \\
& Q_i \in S^n_{dd}, i = 0, \ldots, m
\end{align*}
\]
DSOS programming searches a small subset of nonnegative polynomials set (conservative).
DSOS programming searches a small subset of nonnegative polynomials set (conservative).

To improve the results, we need to increase the search space.

For this, we define “scaled-diagonally-dominant SOS” Polynomials (SDSOS).
Scaled Diagonally Dominant Matrix (sdd)

$Q \in S^n$ is sdd, if there exist a diagonal matrix $D$ with positive diagonal entries, such that $DQD$ is dd.
Scaled Diagonally Dominant Matrix (sdd)

\( Q \in S^n \) is sdd, if there exist a diagonal matrix \( D \) with positive diagonal entries, such that \( DQD \) is dd.

\[
Q = \begin{bmatrix}
1 & 0 & 2 \\
0 & 3 & 0 \\
2 & 0 & 4
\end{bmatrix} \notin dd
\]

1. \( 1 \geq |0| + |2| \) \( \times \)

2. \( 3 \geq |0| + |0| \) \( \checkmark \)

3. \( 4 \geq |2| + |0| \) \( \checkmark \)
Scaled Diagonally Dominant Matrix (sdd)

\( Q \in S^n \) is sdd, if there exist a diagonal matrix \( D \) with positive diagonal entries, such that \( DQD \) is dd.

\[
Q = \begin{bmatrix}
1 & 0 & 2 \\
0 & 3 & 0 \\
2 & 0 & 4
\end{bmatrix} \quad \not\in \text{dd} \\
\begin{align*}
1 & \geq |0| + |2| & \times \\
3 & \geq |0| + |0| & \checkmark \\
4 & \geq |2| + |0| & \checkmark
\end{align*}
\]

\[
D \succ 0 \\
Q \\
D \succ 0
\]

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & \frac{1}{2}
\end{bmatrix} \quad \begin{bmatrix}
1 & 0 & 2 \\
0 & 3 & 0 \\
2 & 0 & 4
\end{bmatrix} \quad \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & \frac{1}{2}
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 1 \\
0 & 3 & 0 \\
1 & 0 & 1
\end{bmatrix} \in \text{dd} \\
\begin{align*}
1 & \geq |0| + |1| & \checkmark \\
3 & \geq |0| + |0| & \checkmark \\
1 & \geq |0| + |1| & \checkmark
\end{align*}
\]

\( Q \) is sdd.
Scaled Diagonally Dominant Matrix (sdd)

$Q \in S^n$ is sdd, if there exist a diagonal matrix $D$ with positive diagonal entries, such that $DQD$ is dd.

$$Q = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 3 & 0 \\ 2 & 0 & 4 \end{bmatrix} \notin dd \quad \begin{array}{l} 1 \geq |0| + |2| \quad \times \\ 3 \geq |0| + |0| \quad \checkmark \\ 4 \geq |2| + |0| \quad \checkmark \end{array}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 \\ 0 & 3 & 0 \\ 2 & 0 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 3 & 0 \\ 1 & 0 & 1 \end{bmatrix} \in dd \quad \begin{array}{l} 1 \geq |0| + |1| \quad \checkmark \\ 3 \geq |0| + |0| \quad \checkmark \\ 1 \geq |0| + |1| \quad \checkmark \end{array}$$

$Q$ is sdd.

$S^n_{dd} \subset S^n_{sdd} \subset S^n_+$

Every dd matrix is sdd matrix with $D = I$
Scaled Diagonally Dominant Matrix (sdd)

To characterize the “sdd” matrices in terms of its element, we use the following result:

\[ Q \in S^n \text{ is sdd if and only if it can be written as} \quad Q = \sum_{i,j=1,...,n,i<j} M^{ij} \]

where, \( M^{ij} \in S^n \)
Scaled Diagonally Dominant Matrix (sdd)

To characterize the “sdd” matrices in terms of its element, we use the following result:

\[ Q \in S^n \text{ is sdd if and only if it can be written as } \quad Q = \sum_{i,j=1,...,n,i<j} M^{ij} \]

where, \( M^{ij} \in S^n \) with zero everywhere except at most for 4 entries

\[(M^{ij})_{ii}, (M^{ij})_{ij}, (M^{ij})_{ji}, (M^{ij})_{jj}\]

which makes the 2 \times 2 matrix

\[
\begin{bmatrix}
(M^{ij})_{ii} & (M^{ij})_{ij} \\
(M^{ij})_{ji} & (M^{ij})_{jj}
\end{bmatrix}
\]

symmetric and positive semidefinite.
Scaled Diagonally Dominant Matrix (sdd)

To characterize the “sdd” matrices in terms of its element, we use the following result:

\[ Q \in S^n \text{ is sdd if and only if it can be written as } Q = \sum_{i,j=1,\ldots,n, i < j} M^{ij} \]

where, \( M^{ij} \in S^n \) with zero everywhere except at most for 4 entries

\[
\begin{pmatrix}
(M^{ij})_{ii}, (M^{ij})_{ij}, (M^{ij})_{ji}, (M^{ij})_{jj}
\end{pmatrix}
\]

which makes the 2 \times 2 matrix symmetric and positive semidefinite.

**Example:**

\[ Q = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 3 & 0 \\ 2 & 0 & 4 \end{bmatrix} \notin \text{dd}, \in \text{sdd} \quad \Rightarrow \quad Q = \sum_{i,j=1,2,3, i < j} M^{ij} = M^{12} + M^{13} + M^{23} \]

\[
\begin{bmatrix}
(M^{12})_{11}, (M^{12})_{12}, (M^{12})_{21}, (M^{12})_{22} \\
(M^{13})_{11}, (M^{13})_{13}, (M^{13})_{31}, (M^{13})_{33} \\
(M^{23})_{22}, (M^{23})_{23}, (M^{23})_{32}, (M^{23})_{33}
\end{bmatrix}
\]

\[ Q = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 3 & 0 \\ 2 & 0 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 0 \end{bmatrix} \]
Scaled Diagonally Dominant Matrix (sdd)

To characterize the “sdd” matrices in terms of its element, we use the following result:

\[ Q \in S^n \text{ is sdd if and only if it can be written as } \quad Q = \sum_{i,j=1,\ldots,n, i < j} M^{ij} \]

where, \( M^{ij} \in S^n \) with zero every where except at most for 4 entries

\[
\begin{pmatrix}
(M^{ij})_{ii}, (M^{ij})_{ij}, (M^{ij})_{ji}, (M^{ij})_{jj}
\end{pmatrix}
\]

which makes the \( 2 \times 2 \) matrix symmetric and positive semidefinite.

Example: \( Q = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 3 & 0 \\ 2 & 0 & 4 \end{bmatrix} \notin dd, \in sdd \quad \Rightarrow \quad Q = \sum_{i,j=1,2,3, i < j} M^{ij} = M^{12} + M^{13} + M^{23} \)
Scaled Diagonally Dominant Matrix (sdd)

\[ S_{dd}^n \subset S_{sdd}^n \subset S_+^n \]

Every dd matrix is sdd matrix with \( D = I \)

Every sdd matrix is sum of psd matrices \( M^{ij} \)

\[
p(x) = B^T(x)QB(x) \quad p(x) = B^T(x)QB(x) \quad p(x) = B^T(x)QB(x)
\]

\[
p(x) \in SOS \quad Q \in S_+^n \quad SDP
\]

\[
p(x) \in DSOS \quad Q \in S_{dd}^n \quad LP
\]

\[
p(x) \in SDSOS \quad Q \in S_{sdd}^n \quad ?
\]
Scaled Diagonally Dominant Matrix (sdd)

\( Q \in S^n \) is sdd if and only if it can be written as

\[
Q = \sum_{i,j = 1, \ldots, n, i < j} M^{ij}
\]

where, \( M^{ij} \in S^n \) with zero everywhere except at most for 4 entries

\[
(M^{ij})_{ii}, (M^{ij})_{ij}, (M^{ij})_{ji}, (M^{ij})_{jj}
\]

which makes the \( 2 \times 2 \) matrix

\[
\begin{bmatrix}
(M^{ij})_{ii} & (M^{ij})_{ij} \\
(M^{ij})_{ji} & (M^{ij})_{jj}
\end{bmatrix}
\]

symmetric and positive semidefinite.
Scaled Diagonally Dominant Matrix (sdd)

$Q \in \mathcal{S}^n$ is sdd if and only if it can be written as

$$Q = \sum_{i,j=1,\ldots,n, i<j} M^i_j$$

where, $M^i_j \in \mathcal{S}^n$ with zero everywhere except at most for 4 entries

$$\begin{bmatrix} (M^i_j)_{ii} & (M^i_j)_{ij} \\ (M^i_j)_{ji} & (M^i_j)_{jj} \end{bmatrix}$$

which makes the $2 \times 2$ matrix symmetric and positive semidefinite.

$$\begin{bmatrix} (M^i_j)_{ii} & (M^i_j)_{ij} \\ (M^i_j)_{ji} & (M^i_j)_{jj} \end{bmatrix} \succeq 0$$

trace(.) $= \lambda_1 + \lambda_2 \geq 0$

det(.) $= \lambda_1 \lambda_2 \geq 0$

1) $(M^i_j)_{ii} + (M^i_j)_{jj} \geq 0$

2) $(M^i_j)_{ji}(M^i_j)_{jj} - (M^i_j)_{jj}(M^i_j)_{ji} \geq 0$
Scaled Diagonally Dominant Matrix (sdd)

$Q \in S^n$ is sdd if and only if it can be written as

$$Q = \sum_{i,j=1,...,n, i<j} M^{ij}$$

where, $M^{ij} \in S^n$ with zero everywhere except at most for 4 entries

$$(M^{ij})_{ii}, (M^{ij})_{ij}, (M^{ij})_{ji}, (M^{ij})_{jj}$$

which makes the $2 \times 2$ matrix

$$\begin{bmatrix}
(M^{ij})_{ii} & (M^{ij})_{ij} \\
(M^{ij})_{ji} & (M^{ij})_{jj}
\end{bmatrix}$$

symmetric and positive semidefinite.

\[\begin{align*}
\text{trace}(.) &= \lambda_1 + \lambda_2 \geq 0 \\
\text{det}(.) &= \lambda_1 \lambda_2 \geq 0 \\
\lambda_1 &\geq 0, \lambda_2 \geq 0
\end{align*}\]

1) $\lambda_1 + \lambda_2 \geq 0$
2) $(M^{ij})_{ii} + (M^{ij})_{jj} \geq 0$

\[\left\| C_i x + d_i \right\|_2 \leq e_i^T x + f_i, \ i = 1, ..., m\]

**Second Order Cone**

$$\left\| \begin{bmatrix}
2(M^{ij})_{ij} \\
(M^{ij})_{ii} - (M^{ij})_{jj}
\end{bmatrix} \right\|_2 \leq (M^{ij})_{ii} + (M^{ij})_{jj}$$

---

Scaled Diagonally Dominant Matrix (sdd)

\( Q \in S^n \) is sdd, if there exist a diagonal matrix \( D \) with positive diagonal entries, such that \( DQD \) is dd.

\[
S^n_{dd} \subset S^n_{sdd} \subset S^n_+
\]

Every dd matrix is sdd matrix with \( D = I \)

Every sdd matrix is sum of psd matrices \( M_{ij} \)

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p(x) \in SOS ) ( p(x) = B^T(x)QB(x) )</td>
<td>PSD Matrix</td>
</tr>
<tr>
<td>( Q \in S^n_+ ) SDP</td>
<td></td>
</tr>
<tr>
<td>( p(x) \in DSOS ) ( p(x) = B^T(x)QB(x) )</td>
<td>LP</td>
</tr>
<tr>
<td>( Q \in S^n_{dd} )</td>
<td></td>
</tr>
<tr>
<td>( p(x) \in SDSOS ) ( p(x) = B^T(x)QB(x) )</td>
<td>Second Order Cone Program (SOCP)</td>
</tr>
<tr>
<td>( Q \in S^n_{sdd} )</td>
<td></td>
</tr>
</tbody>
</table>
Scaled Diagonally Dominant Matrix (sdd)

$Q \in S^n$ is \textit{sdd}, if there exist a diagonal matrix $D$ with positive diagonal entries, such that $DQD$ is \textit{dd}.

\[ S^n_{dd} \subset S^n_{sdd} \subset S^n_+ \]

Every \textit{dd} matrix is \textit{sdd} matrix with $D = I$

Every \textit{sdd} matrix is sum of psd matrices $M_{ij}$

\[
p(x) = B^T(x)QB(x)
\]

$p(x) \in SOS \quad \iff \quad Q \in S^n_+$

SDP

\[
p(x) = B^T(x)QB(x)
\]

$p(x) \in DSOS \quad \iff \quad Q \in S^n_{dd}$

LP

\[
p(x) = B^T(x)QB(x)
\]

$p(x) \in SDSOS \quad \iff \quad Q \in S^n_{sdd}$

Second Order Cone Program (SOCP)

(Appendix I)
Unconstrained optimization

\[
\begin{align*}
\text{minimize} & \quad p(x) \\
\text{subject to} & \quad p(x) - \gamma \geq 0, \quad \forall x \in \mathbb{R}^n
\end{align*}
\]

Constrained optimization

\[
\begin{align*}
\text{maximize} & \quad \gamma \\
\text{subject to} & \quad x \in K = \{x \in \mathbb{R}^n : g_i(x) \geq 0, \ i = 1, \ldots, m\}
\end{align*}
\]

SOS Programming: SOS SDP

\[
\begin{align*}
\text{maximize} & \quad \gamma \\
\text{subject to} & \quad p(x) - \gamma = B^T(x)Q_+B(x) \\
& \quad Q \in S^n_+
\end{align*}
\]

SOS Programming: SOS SDP

\[
\begin{align*}
\text{maximize} & \quad \gamma \\
\text{subject to} & \quad p(x) - \gamma = B^T(x)Q_+B(x) \\
& \quad \sigma_i(x) = B^T_{d_i}(x)Q_iB_{d_i}(x), \ i = 1, \ldots, m \\
& \quad Q_i \in S^n_+, i = 0, \ldots, m
\end{align*}
\]

SDSOS Programming: SOCP

\[
\begin{align*}
\text{maximize} & \quad \gamma \\
\text{subject to} & \quad p(x) - \gamma = B^T(x)Q_{sdd}B(x) \\
& \quad Q \in S^{n}_{sdd}
\end{align*}
\]

SDSOS Programming: SOCP

\[
\begin{align*}
\text{maximize} & \quad \gamma \\
\text{subject to} & \quad p(x) - \gamma - \sum_{i=1}^{m} \sigma_i(x)g_i(x) = B_d(x)Q_0B_d(x) \\
& \quad \sigma_i(x) = B^T_{d_i}(x)Q_iB_{d_i}(x), \ i = 1, \ldots, m \\
& \quad Q_i \in S^{n}_{sdd}, i = 0, \ldots, m
\end{align*}
\]
SPOTT: MATLAB package for DSOS and SDSOS optimization written using the SPOT toolbox.

Link to spotless isos software package:
https://github.com/anirudhamajumdar/spotless/tree/spotless isos

- A. Majumdar, A. A. Ahmadi, R. Tedrake, “Control and verification of high-dimensional systems with DSOS and SDSOS programming”, 53rd IEEE Conference on Decision and Control 2014

Applications:
Control and analyze of high dimensional systems
\[
P^* = \min_{x \in \mathbb{R}^2} \quad 3 + 2x_1 + 2x_2 + 3x_1^2 + 2x_1x_2 + 3x_2^2 + x_1^4 + x_2^4
\]

SDSOS Programming in SPOT

\[
x = \text{msspoly}('x',2);
\]

\[
\text{prog} = \text{spotsosprog};
\]

\[
\text{prog} = \text{prog.withIndeterminate}(x);
\]

\[
p = 3+2*x(1)+2*x(2)+3*x(1)^2+2*x(1)*x(2)+3*x(2)^2+x(1)^4+x(2)^4;
\]

\[
[p,gamma] = \text{prog.newFree}(1);
\]

\[
\text{prog} = \text{prog.withSDSOS}(p-gamma);
\]

\[
\text{sol} = \text{prog.minimize}(-gamma,@\text{spot_mosek});
\]

\[
\text{double(sol.eval(gamma))}
\]

\[
P_{sos}^2 = 2.5074 = P^* \\
P_{sos}^2 \leq P_{sdsos}^2 = 2.0877 \leq P_{sdsos}^2 = 1 \leq P_{sdsos}^2
\]

https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/SDSOS-DSOS/Example_SDSOS_1.m
The problem is to minimize \( P^* = \min_{x \in \mathbb{R}^2} (1 + x_1 x_2)^2 - x_1 x_2 + (1 - x_2)^2 \)

subject to \( x \in K = \{ x \in \mathbb{R}^2 : 3 - 2x_2 - x_1^2 - x_2^2 \geq 0, -x_1 - x_2 - x_1 x_2 \geq 0, 1 + x_1 x_2 \geq 0 \} \)

Variables: \( x_1, x_2 \)

Relaxation order: DSOS/SDSOS Programming

DSOS/SDSOS Programming

\[ d = 1; \]
\[ x = \text{msspoly}('x', 2); \]
\[ \text{prog} = \text{spotsosprog}; \]
\[ \text{prog} = \text{prog}.\text{withIndeterminate}(x); \]
\[ p = (1+x(1)^{2} - x(1)*x(2)+(1-x(2))^{2}; \]
\[ g = \begin{bmatrix} 3-2*x(2) - x(1)^{2} - x(2)^{2}; -x(1) - x(2) - x(1)*x(2); 1 + x(1)*x(2) \end{bmatrix}; \]

\[ \text{[prog, gamma]} = \text{prog}.\text{newFree}(1); \]
\[ \text{mos} = \text{monomials}(x, 0:2*d); \]
\[ \text{[prog, coeffs1]} = \text{prog}.\text{newFree}(	ext{length}(mos)); \]
\[ \text{s1} = \text{coeffs1}' \ast \text{mos}; \]
\[ \text{[prog, coeffs2]} = \text{prog}.\text{newFree}(	ext{length}(mos)); \]
\[ \text{s2} = \text{coeffs2}' \ast \text{mos}; \]
\[ \text{[prog, coeffs3]} = \text{prog}.\text{newFree}(	ext{length}(mos)); \]
\[ \text{s3} = \text{coeffs3}' \ast \text{mos}; \]

\[ \text{prog} = \text{prog}\cdot\text{withSDSOS}(p - \text{gamma} - \begin{bmatrix} s1 & s2 & s3 \end{bmatrix} \ast g); \]
\[ \text{prog} = \text{prog}\cdot\text{withSDSOS}(s1); \]
\[ \text{prog} = \text{prog}\cdot\text{withSDSOS}(s2); \]
\[ \text{prog} = \text{prog}\cdot\text{withSDSOS}(s3); \]

\[ \text{sol} = \text{prog}\cdot\text{minimize}(-\text{gamma}, \text{@spot_mosek}); \]
\[ \text{double}((\text{sol}.\text{eval}((\text{gamma})))) \]

\[ P_{sos}^* = 0.7549 = P^* \]
\[ P_{sdsos}^* = 0.7549 = P_{sos}^* \]
\[ P_{dsos}^* = 0.5 \leq P_{sdsos}^* \]
\[ P_{sdsos}^2 = 0.6585 \quad P_{sdsos}^3 = 0.6891 \quad P_{sdsos}^4 = 0.6935 \quad P_{sdsos}^5 = 0.6937 \quad P_{sdsos}^6 = 0.6937 \]

SOS Polynomials
SDSOS Polynomials
DSOS Polynomials


Fall 2019

https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/SDSOS-DSOS/Example_SDSOS_2.m
P* = \minimize_{x \in \mathbb{R}^2} \frac{1}{3} x_1^6 - \frac{21}{10} x_1^4 + 4x_2 + x_1 x_2 + 4x_2^4 - 4x_2^2 + \frac{3}{2}

\text{subject to } x \in K = \{ x \in \mathbb{R}^2 : -\frac{1}{16} x_1^4 + \frac{1}{4} x_1^3 - \frac{1}{4} x_1^2 - \frac{9}{100} x_2^2 + \frac{29}{400} \geq 0 \}

P_{sos}^* = 0.4684 = P^*

P_{sdsos}^* = 0.3114 \leq P_{sos}^* \quad P_{sdsos}^* = 0.3132 \quad P_{sdsos}^* = 0.3538

P_{dso}^* = -0.0341 \leq P_{dso}^* \quad P_{dso}^* = -0.0061 \quad P_{dso}^* = -0.0353

https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/SDSOS-DSOS/Example_SDSOS_3.m

**Main Benefit:**
SDSOS/DSOS can scale to problems where SOS programming ceases to run due to memory/computation constraints.

Illustrative Example:

\[ \mathbf{P^*} = \min_{x \in \mathbb{R}^n} \quad 5 + \sum_{i=1}^{n} (x_i - 1)^2 \]

\[ p^* = 5, \quad x^* = [1, 1, \ldots, 1]^T \in \mathbb{R}^n \]

Number of variables  | Polynomial of order 2  | Polynomial of order 2  | Polynomial of order 2  | Polynomial of order 2  |
--- | --- | --- | --- |
SOS: Variables:200  | Relaxation Order=1  | time= 286.5458 (s)  | \( p^* = 5 \)  | sdp solver: mosek |
SDSOS: Variables:200  | Relaxation Order=1  | time = 3.6338 (s)  | \( p^* = 5 \)  | sdp solver: mosek |
DSOS: Variables:200  | Relaxation Order=1  | time = 2.6824 (s)  | \( p^* = 5 \)  | sdp solver: mosek |

https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/SDSOS-DSOS/Example_SDSOS_compare_Uncons.m
Bounded Degree SOS

Nonnegative polynomial

\[ p(x) \geq 0, \quad \forall x \in K = \{ x \in \mathbb{R}^n : g_i(x) \geq 0, \ i = 1, \ldots, m \} \]

Putinar’s Positivity Certificate

\[ p(x) = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x) g_i(x) \]

\[ \sigma_i(x) \in \text{SOS}_{2d_i}, \ i = 0, \ldots, m \]

SDP Relaxation

\[ p(x) - \sum_{i=1}^{m} \sigma_i(x) g_i(x) = B_d(x) Q_0 B_d(x) \]

\[ \sigma_i(x) = B_{d_i}^T(x) Q_i B_{d_i}(x), \ i = 1, \ldots, m \]

\[ Q_i \in \mathcal{S}^n_+, \ i = 0, \ldots, m \]
Nonnegative polynomial

\[ p(x) \geq 0, \quad \forall x \in K = \{ x \in \mathbb{R}^n : g_i(x) \geq 0, \ i = 1, \ldots, m \} \]

Putinar’s Positivity Certificate

\[ p(x) = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x)g_i(x) \]

\[ \sigma_i(x) \in SOS_{2d_i}, \ i = 0, \ldots, m \]

Krivine-Stengle’s Positivity Certificate

Let \( K = \{ x \in \mathbb{R}^n : 0 \leq g_i(x) \leq 1, \ i = 1, \ldots, m \} \) \( \text{(normalized polynomials)} \)

SDP Relaxation

\[ p(x) - \sum_{i=1}^{m} \sigma_i(x)g_i(x) = B_d(x)Q_0B_d(x) \]

\[ \sigma_i(x) = B_{d_i}^T(x)Q_iB_{d_i}(x), \ i = 1, \ldots, m \]

\[ Q_i \in S^n_+, i = 0, \ldots, m \]

LP Relaxation
**Nonnegative polynomial**

\[ p(x) \geq 0, \quad \forall x \in \mathbf{K} = \{ x \in \mathbb{R}^n : g_i(x) \geq 0, \ i = 1, \ldots, m \} \]

**Putinar’s Positivity Certificate**

\[
p(x) = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x)g_i(x)
\]
\[
\sigma_i(x) \in \text{SOS}_{2d_i}, \ i = 0, \ldots, m
\]

SDP Relaxation

\[
p(x) - \sum_{i=1}^{m} \sigma_i(x)g_i(x) = B_d(x)Q_0B_d(x)
\]
\[
\sigma_i(x) = B_{d_i}^T(x)Q_iB_{d_i}(x), \ i = 1, \ldots, m
\]
\[
Q_i \in \mathcal{S}_+^n, i = 0, \ldots, m
\]

**Krivine-Stengle’s Positivity Certificate**

Let \( \mathbf{K} = \{ x \in \mathbb{R}^n : 0 \leq g_i(x) \leq 1, \ i = 1, \ldots, m \} \) (normalized polynomials)

\[
p(x) = \sum_{\alpha, \beta \in \mathbb{N}^m} \lambda_{\alpha \beta} g_1^{\alpha_1}(x) \cdots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \cdots (1 - g_m(x))^{\beta_m}
\]

Unknowns: \( \lambda_{\alpha \beta} \) Finitely many Nonnegative scalars

Nonnegative polynomial

\[ p(x) \geq 0, \quad \forall x \in K = \{ x \in \mathbb{R}^n : g_i(x) \geq 0, \; i = 1, \ldots, m \} \]

### Putinar’s Positivity Certificate

\[ p(x) = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x) g_i(x) \]

\[ \sigma_i(x) \in \text{SOS}_{2d_i}, \; i = 0, \ldots, m \]

\[ p(x) - \sum_{i=1}^{m} \sigma_i(x) g_i(x) = B_d(x) Q_0 B_d(x) \]

\[ \sigma_i(x) = B_{d_i}^T(x) Q_{i} B_{d_i}(x), \; i = 1, \ldots, m \]

\[ Q_{i} \in \mathcal{S}_{+}^n, \; i = 0, \ldots, m \]

### Krivine-Stengle’s Positivity Certificate

\[ x \in K \quad p(x) = \sum \lambda_{\alpha_1 \beta_1} g_1^{\alpha_1}(x) \ldots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \ldots (1 - g_m(x))^{\beta_m} \geq 0 \]

\[ x \notin K \quad p(x) = \sum \lambda_{\alpha_1 \beta_1} g_1^{\alpha_1}(x) \ldots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \ldots (1 - g_m(x))^{\beta_m} \geq 0 \]

\[ g_1(x) \leq 0 \; \text{or} \; g_1(x) \geq 1 \]

\[ p(x) \geq 0 \quad \forall x \in K \]

\[ p(x) \geq 0 \quad \text{or} \quad p(x) \leq 0 \]

---

Nonnegative polynomial

\[ p(x) \geq 0, \quad \forall x \in K = \{ x \in \mathbb{R}^n : g_i(x) \geq 0, \ i = 1, ..., m \} \]

Putinar’s Positivity Certificate

\[ p(x) = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x) g_i(x) \]

\[ \sigma_i(x) \in SOS_{2d_i}, \ i = 0, ..., m \]

\[ p(x) - \sum_{i=1}^{m} \sigma_i(x) g_i(x) = B_d(x) Q_0 B_d(x) \]

\[ \sigma_i(x) = B_d^T(x) Q_i B_d(x), \ i = 1, ..., m \]

\[ Q_i \in S^n_+, i = 0, ..., m \]

Krivine-Stengle’s Positivity Certificate

Let \( K = \{ x \in \mathbb{R}^n : 0 \leq g_i(x) \leq 1, \ i = 1, ..., m \} \) (normalized polynomials)

\[ p(x) = \sum_{\alpha, \beta \in \mathbb{N}^m} \lambda_{\alpha \beta} g_1^{\alpha_1}(x) ... g_m^{\alpha_m}(x) (1 - g_1(x))^{\beta_1} ... (1 - g_m(x))^{\beta_m} \]

Unknowns: \( \lambda_{\alpha \beta} \) Finitely many Nonnegative scalars

- Determining if \( p(x) \geq 0, \forall x \in K \) leads to a linear optimization feasibility problem.

\[ \begin{align*}
\mathbf{P}^* = & \minimize_{x \in \mathbb{R}^n} \quad p(x) \\
\text{subject to} \quad & x \in \mathbf{K} = \{x \in \mathbb{R}^n : g_i(x) \geq 0, \ i = 1, \ldots, m\}
\end{align*} \]

\[ \begin{align*}
\maximize_{\gamma \in \mathbb{R}} \quad & \gamma \\
\text{subject to} \quad & p(x) - \gamma \geq 0, \ \forall x \in \mathbf{K}
\end{align*} \]

**SDP Relaxation**

\[ \begin{align*}
\maximize_{\gamma, \mathbf{Q}_i} \quad & \gamma \\
\text{subject to} \quad & p(x) - \gamma - \sum_{i=1}^{m} \sigma_i(x) g_i(x) = B_d(x)Q_0B_d(x) \\
& \sigma_i(x) = B_{d_i}(x)Q_iB_{d_i}(x), \ i = 1, \ldots, m \\
& Q_i \in \mathbb{S}^n_+, i = 0, \ldots, m
\end{align*} \]

**LP Relaxation**

Let \( d \in \mathbb{N} \)

\[ \begin{align*}
\mathbf{P}_{L}^{*d} = & \maximize_{\gamma, \lambda_{\alpha,\beta} \geq 0} \quad \gamma \\
\text{subject to} \quad & p(x) - \gamma = \sum_{\forall \alpha, \beta \in \mathbb{N}^m} \lambda_{\alpha,\beta} g_1^{\alpha_1}(x) \cdots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \cdots (1 - g_m(x))^{\beta_m} \\
& \sum_{j=1}^{m} \alpha_j + \beta_j \leq d
\end{align*} \]

**Theorem:** Let \( \mathbf{K} \) be compact (Archimedean). \( \mathbf{P}^{*d} \leq \mathbf{P}^{*d+1} \) \( \lim_{d \to \infty} \mathbf{P}^{*d} = \mathbf{P}^* \)

**Theorem 5.10.** Jean Bernard Lasserre, “Moments, Positive Polynomials and Their Applications” Imperial College Press Optimization Series, V. 1, 2009.
LP-relaxations suffer from several serious theoretical and practical drawbacks:

- The LPs of the hierarchy are numerically **ill-conditioned**.
  - It involves products of arbitrary powers of the $g_i(x)$’s and $(1 - g_i(x))$’s.
  - In particular, the presence of large coefficients is source of ill-conditioning and numerical instability.

- The sequence of the associated optimal values converges to the global optimum only **asymptotically** and **not in finitely** many steps. (*Appendix II*)

- Finite convergence even does not hold for **convex optimizations**. (In standard SOS finite convergence takes place for SOS-convex problems)
LP-relaxations suffer from several serious theoretical and practical drawbacks:

- The LPs of the hierarchy are numerically ill-conditioned.
  - It involves products of arbitrary powers of the $g_i(x)$’s and $(1 - g_i(x))$’s.
  - In particular, the presence of large coefficients is source of ill-conditioning and numerical instability.

- The sequence of the associated optimal values converges to the global optimum only asymptotically and not in finitely many steps. (Appendix II)

- Finite convergence even does not hold for convex optimizations. (In standard SOS finite convergence takes place for SOS-convex problems)

**Bounded Degree SOS (BSOS):**

Hierarchy of convex relaxations which combines some of the advantages of the SOS and LP hierarchies.
Bounded Degree SOS (BSOS):
Hierarchy of convex relaxations which **combines** some of the advantages of the SOS- and LP- hierarchies.

- **SOS Relaxation**
  \[
  p(x) = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x) g_i(x) \\
  \sigma_0(x) \in \text{SOS}_{2d} \\
  \sigma_i(x) \in \text{SOS}_{2d_i}, \ i = 1, \ldots, m
  \]

- **LP Relaxation**
  \[
  p(x) = \sum_{\substack{i \forall \alpha, \beta \in \mathbb{N}^m \sum_{j=1}^{m} \alpha_j + \beta_j \leq i}} \lambda_{\alpha \beta} g_1^{\alpha_1}(x) \cdots g_m^{\alpha_m}(x) (1 - g_1(x))^{\beta_1} \cdots (1 - g_m(x))^{\beta_m}
  \]

- **BSOS Relaxation**
  \[
  p(x) = \sigma_0(x) + \sum_{\substack{i \forall \alpha, \beta \in \mathbb{N}^m \sum_{j=1}^{m} \alpha_j + \beta_j \leq d}} \lambda_{\alpha \beta} g_1^{\alpha_1}(x) \cdots g_m^{\alpha_m}(x) (1 - g_1(x))^{\beta_1} \cdots (1 - g_m(x))^{\beta_m}
  \]
  \[
  \sigma_0(x) \in \text{SOS}_{2k} \\
  k \in \mathbb{N}: \text{Degree of SOS polynomial} \\
  d \in \mathbb{N}: \text{degree of LP representation}
  \]

\[ \mathbf{P}^* = \text{minimize} \quad p(x) \]
\[ \text{subject to} \quad x \in K = \{ x \in \mathbb{R}^n : g_i(x) \geq 0, \ i = 1, \ldots, m \} \]

\[ \text{maximize} \quad \gamma \]
\[ \text{subject to} \quad p(x) - \gamma - \sum_{i=1}^{m} \sigma_i(x) g_i(x) = B_d(x) Q_0 B_d(x) \]
\[ \sigma_i(x) = B_{d_i}^T(x) Q_i B_{d_i}(x), \ i = 1, \ldots, m \]
\[ Q_i \in S^m_+, i = 0, \ldots, m \]

\[ \mathbf{P}_{s^j}^* = \text{maximize} \quad \gamma \]
\[ \text{subject to} \quad p(x) - \gamma = \sum_{\alpha, \beta} \lambda_{\alpha, \beta} g_1^{\alpha_1}(x) \cdots g_m^{\alpha_m}(x) (1 - g_1(x))^{\beta_1} \cdots (1 - g_m(x))^{\beta_m} = B_k(x) Q_0 B_k(x) \]
\[ Q_0 \in S^m_+ \]

• **Theorem:** Let \( k \in \mathbb{N} \) be fixed. \( \mathbf{P}_{d}^* \leq \mathbf{P}_{d+1}^* \)

\[ \lim_{d \to \infty} \mathbf{P}_{d}^* = \mathbf{P}^* \]

➢ **Finite convergence (Like standard SOS)** (Finite convergence condition: Rank condition of the dual (moment) problem) \( \text{(Appendix III)} \)

➢ Unlike standard SOS, the size of SDP is fixed \( \binom{n+k}{n} \)

* Section 1.1, Jean B. Lasserre, Kim-Chuan Toh, Shouguang Yang, "A bounded degree SOS hierarchy for polynomial optimization", EURO Journal on Computational Optimization March 2017, Volume 5, Issue 1–2, pp 87–117
Example 1

\[(P_1)\]
\[
f = x_1^3 - x_2^3 + x_3^3 - x_4^3 + x_5 - x_2
\]
\[\text{s.t.}\]
\[0 \leq g_1 = 2x_1^2 + 3x_2^2 + 2x_1x_2 + 3x_3^2 + 3x_4^2 + 2x_3x_4 \leq 1
\]
\[0 \leq g_2 = 3x_1^2 + 2x_2^2 - 4x_1x_2 + 3x_3^2 + 2x_4^2 - 4x_3x_4 \leq 1
\]
\[0 \leq g_3 = x_1^2 + 6x_2^2 - 4x_1x_2 + x_3^2 + 6x_4^2 - 4x_3x_4 \leq 1
\]
\[0 \leq g_4 = x_1^2 + 3x_2^2 - 3x_1x_2 + x_3^2 + 4x_4^2 - 3x_3x_4 \leq 1
\]
\[0 \leq g_5 = 2x_1^2 + 5x_2^2 + 3x_1x_2 + 2x_3^2 + 5x_4^2 + 3x_3x_4 \leq 1
\]
\[0 \leq x.
\]

\[\mathbb{P}^{*}_{d=1} = -0.57491 = \mathbb{P}^{*}
\]

Example 2

\[(P_2)\]
\[
f = x_1^2x_2^3 + x_1^3x_2^4 - x_1^4x_2^2
\]
\[\text{s.t.}\]
\[0 \leq g_1 = x_1^2 + x_2^3 \leq 1
\]
\[0 \leq g_2 = 3x_1^2 + 2x_2^2 - 4x_1x_2 \leq 1
\]
\[0 \leq g_3 = x_1^2 + 6x_2^2 - 4x_1x_2 + 2.5 \leq 1
\]
\[0 \leq g_4 = x_1^2 + 3x_2^2 \leq 1
\]
\[0 \leq g_5 = x_1^2 + x_2^3 \leq 1
\]
\[0 \leq x_1, \ 0 \leq x_2.
\]

\[\mathbb{P}^{*} = -0.037037
\]

Fixed size of SDP

\[k = 3\]
\[\mathbb{P}^{*}_{d=1}^{k=3} = -0.041855 \quad \mathbb{P}^{*}_{d=2}^{k=3} = -0.037139 \quad \mathbb{P}^{*}_{d=3}^{k=3} = -0.037087 \quad \mathbb{P}^{*}_{d=4}^{k=3} = -0.037073 \quad \mathbb{P}^{*}_{d=5}^{k=3} = -0.037046
\]

\[k = 4\]
\[\mathbb{P}^{*}_{d=1}^{k=4} = -0.038596 \quad \mathbb{P}^{*}_{d=2}^{k=4} = -0.037046 \quad \mathbb{P}^{*}_{d=3}^{k=4} = -0.037040 \quad \mathbb{P}^{*}_{d=4}^{k=4} = -0.037038 \quad \mathbb{P}^{*}_{d=5}^{k=4} = -0.037037
\]

More examples: [https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/Bounded_Degree_SOS/BSOS_Example1.m](https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/Bounded_Degree_SOS/BSOS_Example1.m)


Code: [https://github.com/tweisser/Sparse_BSOS](https://github.com/tweisser/Sparse_BSOS)

Topics

1) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS)
   Modified SOS optimization that results in LP and Second order cone program

2) Bounded degree SOS (BSOS)
   Modified SOS optimization that results in smaller SDP’s.

3) Sparse Sum-of-Squares Optimization (SSOS)
   Takes advantage of sparsity of the original problem to generate smaller SDP.

4) Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)
   Combination of 2 and 3
Sparse SOS


➢ Take advantage of structure (sparsity) of the problem to solve smaller SDP
Take advantage of structure (sparsity) of the problem to solve smaller SDP

1) PSD Constraint obtained from SOS/Moment Relaxation.

- (Under some conditions) We can replace Constraint of the form $Q \succeq 0$ by PSD constraints of set of smaller matrices.

Example:

$$Q = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 0.5 & 1 \\ 0 & 1 & 2 \end{bmatrix} \Rightarrow Q \text{ is PSD because:} \begin{bmatrix} 2 & 1 \\ 1 & 0.5 \end{bmatrix} \succeq 0 , \begin{bmatrix} 0.5 & 1 \\ 1 & 2 \end{bmatrix} \succeq 0$$
- Take advantage of structure (sparsity) of the problem to solve smaller SDP

1) PSD Constraint obtained form SOS/Moment Relaxation.

   - (Under some conditions) We can replace Constraint of the form $Q \succeq 0$ by PSD constraints of set of smaller matrices.

   **Example:**
   
   $$Q = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0.5 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 1 \\ 0 & 1 & 2 \end{bmatrix} \Rightarrow Q \text{ is PSD because: } \begin{bmatrix} 2 & 1 \\ 1 & 0.5 \end{bmatrix} \succeq 0, \begin{bmatrix} 0.5 & 1 \\ 1 & 2 \end{bmatrix} \succeq 0$$

2) SOS relaxation of nonnegative Polynomials

   - (Under some conditions) We can replace constraint of $p(x) \in SOS$ by SOS constraints of low dimensional polynomials.

   **Example:**
   
   $$p(x_1, x_2, x_3) = 2(1 + x_1 + x_3 + x_1^2 + x_1 x_2 + x_2^2 + x_2 x_3 + x_3^2) \Rightarrow p(x_1, x_2, x_3) = p_1(x_1, x_2) + p_2(x_2, x_3)$$
   
   $$p_1(x_1, x_2) = (1 + x_1)^2 + (x_1 + x_2)^2$$
   $$p_2(x_2, x_3) = (1 + x_3^2)^2 + (x_2 + x_3)^2$$

   Polynomial $p(x_1, x_2, x_3)$ is SOS because $p_1(x_1, x_2)$ and $p_2(x_2, x_3)$ are SOS.
Sparse Polynomials

Polynomial:

\[ p(x) : \mathbb{R}^n \rightarrow \mathbb{R} \quad p(x) = \sum_{\alpha} p_{\alpha} x^{\alpha} \]

number of coefficients \( \binom{n+d}{n} = \frac{(n+d)!}{n!d!} \)

- Fully dense polynomial: Polynomial is fully dense if all the coefficients are nonzero
Sparse Polynomials

Polynomial: \( p(x) : \mathbb{R}^n \rightarrow \mathbb{R} \)

\[ p(x) = \sum_{\alpha} p_{\alpha} x^\alpha \]

Number of coefficients \( \binom{n+d}{n} = \frac{(n+d)!}{n!d!} \)

- **Fully dense polynomial**: Polynomial is fully dense if all the coefficients are nonzero.

- **Sparse polynomial**: Polynomial is sparse if the number of nonzero coefficients is much smaller than the number of the total coefficients.

Example: Sparse Polynomial

\[ p(x_1, x_2) = 0.56 + 0.5x_1 + 2x_2^2 + 0.75x_1^3x_2^2 \]

Number of nonzero coefficients: 4

Number of all coefficients: \( \binom{2+5}{2} = \binom{7}{2} = \frac{7!}{2!5!} = 21 \)
Sparse Polynomials

Polynomial: \( p(x) : \mathbb{R}^n \rightarrow \mathbb{R} \)
\[ p(x) = \sum_{\alpha} p_{\alpha} x^\alpha \]
number of coefficients \( \binom{n+d}{n} = \frac{(n+d)!}{n!d!} \)

- **Fully dense polynomial:** Polynomial is fully dense if all the coefficients are nonzero.

- **Sparse polynomial:** Polynomial is sparse if the number of nonzero coefficients is much smaller than the number of the total coefficients.

Example: Sparse Polynomial
\[ p(x_1, x_2) = 0.56 + 0.5x_1 + 2x_2^2 + 0.75x_1^3x_2^2 \]

Number of nonzero coefficients: 4
Number of all coefficients: \( \binom{2+5}{2} = \frac{(7)!}{2!5!} = 21 \)

- **Correlative Sparsity:** It describes coupling between the variables \( x_1, \ldots, x_n \) of a polynomial \( p(x) : \mathbb{R}^n \rightarrow \mathbb{R} \).
  
  - Variables \( x_i \) and \( x_j \) are coupled if they appear simultaneously in a monomial of the polynomial.
**Sparse Polynomials**

Polynomial: \( p(x) : \mathbb{R}^n \rightarrow \mathbb{R} \)  

\[
p(x) = \sum_{\alpha} p_{\alpha} x^{\alpha}
\]

Number of coefficients  

\[
\binom{n+d}{n} = \frac{(n+d)!}{n!d!}
\]

- **Fully dense polynomial**: Polynomial is fully dense if all the coefficients are nonzero.

- **Sparse polynomial**: Polynomial is sparse if the number of nonzero coefficients is much smaller than the number of the total coefficients.

Example: Sparse Polynomial  

\[
p(x_1, x_2) = 0.56 + 0.5x_1 + 2x_2^2 + 0.75x_1^3x_2^2
\]

Number of nonzero coefficients: 4  

Number of all coefficients:  \( \binom{2+5}{2} = \frac{(7)!}{2!5!} = 21 \)

- **Correlative Sparsity**: It describes coupling between the variables \( x_1, \ldots, x_n \) of a polynomial \( p(x) : \mathbb{R}^n \rightarrow \mathbb{R} \).

  - Variables \( x_i \) and \( x_j \) are coupled if they appear simultaneously in a monomial of the polynomial.

Example:  

\[
p(x_1, x_2, x_3, x_4) = 0.56 + 0.5x_1 + 2x_1x_2^2 + 0.75x_3^3x_4^2
\]

Coupled variables: \( (x_1, x_2), (x_3, x_4) \)

Missing Coupled variables: \( (x_1, x_3), (x_1, x_4), (x_2, x_3), (x_2, x_4) \)
Sparse Polynomials

Polynomial: \( p(x) : \mathbb{R}^n \rightarrow \mathbb{R} \quad p(x) = \sum_{\alpha} p_{\alpha} x^\alpha \) number of coefficients \( \binom{n+d}{n} = \frac{(n+d)!}{n!d!} \)

- **Fully dense polynomial:** Polynomial is fully dense if all the coefficients are nonzero.

- **Sparse polynomial:** Polynomial is sparse if the number of nonzero coefficients is much smaller than the number of the total coefficients.

Example: Sparce Polynomial \( p(x_1, x_2) = 0.56 + 0.5x_1 + 2x_2^2 + 0.75x_1^3x_2^2 \) Number of nonzero coefficients: 4 Number of all coefficients: \( \binom{2+5}{2} = \frac{(7)!}{2!5!} = 21 \)

- **Correlative Sparsity:** It describes coupling between the variables \( x_1, \ldots, x_n \) of a polynomial \( p(x) : \mathbb{R}^n \rightarrow \mathbb{R} \)

  - Variables \( x_i \) and \( x_j \) are coupled if they appear simultaneously in a monomial of the polynomial.

Example: \( p(x_1, x_2, x_3, x_4) = 0.56 + 0.5x_1 + 2x_1x_2^2 + 0.75x_3^3x_4^2 \) Coupled variables: \( (x_1, x_2), (x_3, x_4) \)

  - Number of all possible coupling between variables \( x_1, \ldots, x_n \) : \( \binom{n}{2} \)

  - Missing Coupled variables: \( (x_1, x_3), (x_1, x_4), (x_2, x_3), (x_2, x_4) \)

- Polynomial has **correlative sparsity** if the number of coupled variables is much smaller than the Number of all possible coupling.
Sparse Polynomials

- **Sparse polynomial**: Polynomial is sparse if the number of nonzero coefficients is much smaller than the number of the total coefficients.

- **Correlative Sparsity**: Polynomial has correlative sparsity if the number of coupled variables is much smaller than the Number of all possible coupling
Sparse Polynomials

- **Sparse polynomial**: Polynomial is **sparse** if the number of nonzero coefficients is much smaller than the number of the total coefficients.

- **Correlative Sparsity**: Polynomial has **correlative sparsity** if the number of coupled variables is much smaller than the Number of all possible coupling

  - Correlative sparsity is a special case of the sparsity.
  
  - Correlative sparsity implies the sparsity, but the converse is not necessarily true.

  \[ p(x_1, x_2, x_3, x_4) = x_1x_2 + x_1x_3 + x_1^3x_4 + x_2x_3 + x_2x_4 + x_3x_4^{10} \]

  Number of nonzero coefficients: 6  
  Number of all coefficients: \( \binom{4+10}{4} = \frac{(14)!}{4!10!} = 1001 \)

  **Sparse Polynomial**  
  With NO correlative sparsity
(Under some conditions) Constraint of the form $p(x) \in SOS$ can be replaced by SOS constraints of low dimensional polynomials.

\[
p(x) \in SOS \iff p(x) = \sum_{k} p_k(X_k) \quad p_k(X_k) \in SOS
\]

$X_k$: Coupled set variables of $p(x)$

(Under some conditions) Constraint of the form $p(x) \in SOS$ can be replaced by SOS constraints of low dimensional polynomials.

$$p(x) \in SOS \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS$$

$X_k$: Coupled set variables of $p(x)$

$$p(x) = B^T(x)QB(x) \quad Q \in S_+^n$$

$$p(x) = \sum_k z_k^T(x)Q_k z_k(x) \quad Q_k \in S_+^{C_k} \quad C_k < n$$

(Under some conditions) Constraint of the form $p(x) \in SOS$ can be replaced by SOS constraints of low dimensional polynomials.

$$p(x) \in SOS \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS$$

$X_k$: Coupled set variables of $p(x)$

$$p(x) = B^T(x)Q B(x) \quad Q \in S^n_+$$

$$p(x) = \sum_k z_k^T(x)Q_k z_k(x) \quad Q_k \in S^C_k \quad C_k < n$$

Example:

$$p(x_1, x_2, x_3) = 2x_1^2 + 2x_1x_2 + x_2^2 + 2x_2x_3 + 2x_3^2$$

$$p(x) \in SOS$$

$$p(x_1, x_2, x_3) = p_1(x_1, x_2) + p_2(x_2, x_3)$$

$$p_1(x_1, x_2) = (\sqrt{2}x_1 + \sqrt{0.5}x_2)^2 \in SOS$$

$$p_2(x_2, x_3) = (\sqrt{0.5}x_2 + \sqrt{2}x_3)^2 \in SOS$$
(Under some conditions) Constraint of the form $p(x) \in SOS$ can be replaced by SOS constraints of low dimensional polynomials.

\[
p(x) \in SOS \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS
\]

$X_k$: Coupled set variables of $p(x)$

\[
p(x) = B^T(x)QB(x) \iff p(x) = \sum_k z_k^T(x)Q_kz_k(x) \quad Q_k \in S^C_k \quad C_k < n
\]

\[Q \in S^n_+\]


**Example:**

\[
p(x_1, x_2, x_3) = 2x_1^2 + 2x_1x_2 + x_2^2 + 2x_2x_3 + 2x_3^2
\]

$p(x) \in SOS \iff p(x_1, x_2, x_3) = p_1(x_1, x_2) + p_2(x_2, x_3)$

\[
p_1(x_1, x_2) = (\sqrt{2}x_1 + \sqrt{0.5}x_2)^2 \in SOS
\]

\[
p_2(x_2, x_3) = (\sqrt{0.5}x_2 + \sqrt{2}x_3)^2 \in SOS
\]
(Under some conditions) Constraint of the form \( p(x) \in SOS \) can be replaced by SOS constraints of low dimensional polynomials.

\[
p(x) \in SOS \quad \iff \quad p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS
\]

\( X_k \): Coupled set variables of \( p(x) \)

\[
p(x) = B^T(x) Q B(x) \quad Q \in S_+^n
\]

\[
p(x) = \sum_k z_k^T(x) Q_k z_k(x) \quad Q_k \in S_+^{C_k} \quad C_k < n
\]

(Under some conditions) Constraint of the form \( p(x) \in SOS \) can be replaced by SOS constraints of low dimensional polynomials.

\[
p(x) \in SOS \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS
\]

\( X_k \): Coupled set variables of \( p(x) \)

\[
p(x) = B^T(x)QB(x)
\]

\( Q \in S^n_+ \)

\[
p(x) = \sum_k z_k^T(x)Q_k z_k(x)
\]

\( Q_k \in S^{C_k}_+ \quad C_k < n \)


(Under some conditions) Constraint of the form \( X \succeq 0 \) can be replaced by PSD constraints of smaller matrices \( X_k \succeq 0 \)

\[
X \succeq 0 \iff X = \sum_k E_k^T X_k E_k
\]

\( n \times n \) matrix

\[
X_k \succeq 0 \\
C_k \times C_k \text{ matirx} \\
C_k \times n \text{ constant matrix}
\]


(Under some conditions) Constraint of the form $p(x) \in SOS$ can be replaced by SOS constraints of low dimensional polynomials.

$$p(x) \in SOS \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS$$

$X_k$: Coupled set variables of $p(x)$

$$p(x) = B^T(x) QB(x) \iff p(x) = \sum_k z_k^T(x) Q_k z_k(x) \quad Q_k \in S_+^{C_k} \quad C_k < n$$

Example:

$X = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad \iff \quad \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 0.5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & 1 \\ 1 & 2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$$X \succcurlyeq 0 \iff X = \sum_k E_k^T X_k E_k \quad X_k \succcurlyeq 0 \quad C_k < n$$

(Under some conditions) Constraint of the form \( p(x) \in SOS \) can be replaced by SOS constraints of low dimensional polynomials.

\[
p(x) \in SOS \quad \text{If and only if} \quad p(x) = \sum_k p_k(X_k) \quad \text{if and only if} \quad X_k \text{: Coupled set variables of } p(x)
\]

\[
p(x) = B^T(x)QB(x) \quad Q \in S_+^n
\]

\[
p(x) = \sum_k z_k^T(x)Q_kz_k(x) \quad Q_k \in S_+^{C_k} \quad C_k < n
\]


(Under some conditions) Constraint of the form \( X \succeq 0 \) can be replaced by PSD constraints of smaller matrices \( X_k \succeq 0 \)

\[
X \succeq 0 \quad \text{If and only if} \quad X = \sum_k E_k^T X_k E_k \quad X_k \succeq 0 \quad C_k < n
\]


(Under some conditions) Constraint of the form $p(x) \in SOS$ can be replaced by SOS constraints of low dimensional polynomials.

$$p(x) \in SOS \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS$$

$X_k$: Coupled set variables of $p(x)$

$$p(x) = B^T(x)QB(x)$$

$Q \in S_+^n$

$$p(x) = \sum_k \tilde{z}_k^T(x)Q_k\tilde{z}_k(x)$$

$Q_k \in S_+^{C_k} \quad C_k < n$


(Under some conditions) Constraint of the form $X \succeq 0$ can be replaced by PSD constraints of smaller matrices $X_k \succeq 0$

$$X \succeq 0 \iff X = \sum_k E_k^T X_k E_k$$

$n \times n$ matirx

$X_k \succeq 0 \quad C_k < n$


Results rely on sparsity pattern of polynomials and Matrices and its graph representation, and Chordality of sparsity graph (the classical theory of graph and cliques).
Undirected Graph

- Undirected graph $\mathcal{G}$
- $\mathcal{V}$ Set of nodes of the graph
- $\mathcal{E}$ Set of edges of the graph
**Undirected Graph**

- We use undirected graph to represent polynomials and symmetric matrices.

\[ p(x_1, x_2, x_3) = 1 + x_1 + x_3 + x_1^2 + x_1 x_2 + x_1^2 + x_2 x_3 + x_3^2 \]

Coupled variables: \((x_1, x_2), (x_2, x_3)\)

Edges between coupled variables

\[ X = \begin{bmatrix} X_{11} & X_{12} & 0 & 0 \\ X_{12} & X_{22} & X_{23} & X_{24} \\ 0 & X_{23} & X_{33} & X_{34} \\ 0 & X_{24} & X_{34} & X_{44} \end{bmatrix} \]

Edges: Nonzero entries of matrix

- **Edges**
  - Nonzero entries of matrix

- **Sparsity pattern**
  - Polynomial
  - Matrix
Undirected Graph

Undirected graph $G$

\( \mathcal{V} \) Set of nodes of the graph

\( \mathcal{E} \) Set of edges of the graph

**Cycle:** A cycle of length $k$ in a undirected graph is a sequence of nodes \((v_1, v_2, ... , v_k)\) such that \((v_i, v_{i+1}) i = 1, ..., k - 1\) and \((v_1, v_k)\) are the edges.
**Undirected Graph**

- **Undirected graph** $G$
  - $V$ Set of nodes of the graph
  - $E$ Set of edges of the graph

**Cycle:** A cycle of length $k$ in an undirected graph is a sequence of nodes $(v_1, v_2, ..., v_k)$ such that $(v_i, v_{i+1})$ for $i = 1, ..., k-1$ and $(v_1, v_k)$ are the edges.

**Chord:** is an edge that connects 2 nonadjacent nodes in a cycle.

![Cycle of length 3](cycle_of_length_3.png)

![Cycle of length 4](cycle_of_length_4.png)

![Cycle of length 5 with 2 chords](cycle_of_length_5_with_2_chords.png)
**Undirected Graph** ➢ Undirected graph $\mathcal{G}$  
\[ V \quad \text{Set of nodes of the graph} \quad E \quad \text{Set of edges of the graph} \]

**Cycle:** A cycle of length $k$ in an undirected graph is a sequence of nodes $(v_1, v_2, \ldots, v_k)$ such that $(v_i, v_{i+1})$ for $i = 1, \ldots, k-1$ and $(v_1, v_k)$ are the edges.

**Chord:** is an edge that connects 2 nonadjacent nodes in a cycle.

**Chordal Graph:** An undirected graph is chordal if every cycle of the length $k \geq 4$ has a chord, (if there are no cycles of length $\geq 4$)
Undirected Graph $\mathcal{G}$

Set of nodes of the graph $\mathcal{V}$

Set of edges of the graph $\mathcal{E}$

**Cycle:** A cycle of length $k$ in a undirected graph is a sequence of nodes $(v_1, v_2, \ldots, v_k)$ such that $(v_i, v_{i+1})$ for $i = 1, \ldots, k-1$ and $(v_1, v_k)$ are the edges.

**Chord:** is an edge that connects 2 nonadjacent nodes in a cycle.

**Chordal Graph:** An undirected graph is chordal if every cycle of the length $k \geq 4$ has a chord, (if there are no cycles of length $\geq 4$)

**Clique:** a clique of a graph is a subset of nodes that construct a complete graph (i.e. each node in the clique is connected to all the nodes in the clique.)
**Undirected Graph**  \( G \)  
\[ V \]  Set of nodes of the graph  
\[ E \]  Set of edges of the graph

**Cycle:** A cycle of length \( k \) in a undirected graph is a sequence of nodes \( (v_1, v_2, \ldots, v_k) \) such that \( (v_i, v_{i+1}) \) \( i = 1, \ldots, k - 1 \) and \( (v_1, v_k) \) are the edges.

**Chord:** is an edge that connects 2 nonadjacent nodes in a cycle.

**Chordal Graph:** An undirected graph is chordal if every cycle of the length \( k \geq 4 \) has a chord, (if there are no cycles of length \( \geq 4 \))

**Clique:** a clique of a graph is a subset of nodes that construct a complete graph (i.e. each node in the clique is connected to all the nodes in the clique.)

**Maximal Clique:** a clique is maximal if it is not a subset of another clique.
Theorem

Let $G(V,E)$ be a chordal graph\(^1\) with maximal cliques \(\{C_1, C_2, \ldots, C_l\}\). Then, Matrix $X \in S^n$ with sparsity pattern $G(V,E)$ is PSD if and only if there exist PSD matrices $X_k \in S^{|C_k|}$ \(\succeq 0\).

\[
X \succeq 0 \quad \text{If and only if} \quad X = \sum_k E_{C_k}^T X_k E_{C_k} \quad X_k \in S^{|C_k|} \succeq 0 \quad \text{Matrices constructed from the maximal Cliques}
\]

\(\Rightarrow\) Constraint of the form $X \succeq 0$ can be replaced by PSD constraints of smaller matrices $X_k \succeq 0$.

\(\Rightarrow\) Number of the nodes in maximal Cliques

Example:

\[
X = \begin{bmatrix}
2 & 1 & 0 \\
1 & 2 & 1 \\
0 & 1 & 2
\end{bmatrix}
\]

\(X \succeq 0\)

---

Let \( G(\mathcal{V}, \mathcal{E}) \) be a chordal graph\(^1\) with maximal cliques \( \{C_1, C_2, \ldots, C_l\} \). Then, Matrix \( X \in S^n \) with sparsity pattern \( G(\mathcal{V}, \mathcal{E}) \) is PSD if and only if there exist PSD matrices \( X_k \in S^{|C_k|} \geq 0 \) such that:

\[
X \succeq 0 \iff X = \sum_k E_{C_k}^T X_k E_{C_k} \quad \text{with sparsity pattern } |C_k| < n
\]

Constraint of the form \( X \succeq 0 \) can be replaced by PSD constraints of smaller matrices \( X_k \succeq 0 \).

**Example:**

\[
X = \begin{bmatrix}
2 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 2
\end{bmatrix}
\]

\[ X \succeq 0 \]

\[
\exists \quad X_1 \in S^{|C_1|} \succeq 0 \quad X_2 \in S^{|C_2|} \succeq 0 \quad \iff \quad X \succeq 0
\]

\[
X = \sum_k E_{C_k}^T X_k E_{C_k}
\]

Theorem

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a chordal graph\(^1\) with maximal cliques $\{C_1, C_2, \ldots, C_t\}$. Then, Matrix $X \in S^n$ with sparsity pattern $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is PSD if and only if there exist PSD matrices $X_k \in S^{|C_k|}$ for all $k$.

\[ X \succeq 0 \iff X = \sum_k E_{C_k}^T X_k E_{C_k} \]

Constraint of the form $X \succeq 0$ can be replaced by PSD constraints of smaller matrices $X_k \succeq 0$.

Example:

\[
X = \begin{bmatrix}
2 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 2
\end{bmatrix}
\quad \iff \quad
X = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 1 \\
0 & 0 & 0
\end{bmatrix} +
\begin{bmatrix}
2 & 1 & 0 \\
1 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix} +
\begin{bmatrix}
0.5 & 1 & 0 \\
1 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix}
\]

\[
X_1 \succeq 0 \\
X_2 \succeq 0
\]

$X \succeq 0 \iff X_1 \succeq 0 \land X_2 \succeq 0$

---

Theorem

Let $G(V, E)$ be a **chordal graph**\(^1\) with maximal cliques $\{C_1, C_2, ..., C_l\}$. Then, Matrix $X \in S^n$ with sparsity pattern $G(V, E)$ is PSD if and only if there exist PSD matrices $X_k \in S^{|C_k|}$ such that:

$$X = \sum_k E_{C_k}^T X_k E_{C_k}, \quad X_k \in S^{|C_k|}, \quad |C_k| < n$$

Constraint of the form $X \succcurlyeq 0$ can be replaced by PSD constraints of smaller matrices $X_k \succcurlyeq 0$.

**Example:**

$$X = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad \succcurlyeq 0$$

$$X = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0.5 & 1 \\ 1 & 2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$X_1 \succcurlyeq 0$, $X_2 \succcurlyeq 0$

**SDP**

\[
\begin{align*}
& \text{minimize} \quad C \cdot X \\
& \text{subject to} \quad A_i \cdot X = b_i \quad i = 1, \ldots, m. \\
& \quad X \succeq 0. \quad X \in S^n
\end{align*}
\]

Sparsity pattern of matrix $X$ : Chordal graph $G(\mathcal{V}, \mathcal{E})$

\[
\begin{align*}
& \text{minimize} \quad C \cdot X \\
& \text{subject to} \quad A_i \cdot \left( \sum_k E_{C_k}^T X_k E_{C_k} \right) = b_i \quad i = 1, \ldots, m. \\
& \quad X_k \succeq 0, \quad k = 1, 2, \ldots \quad X_k \in S^{|C_k|}
\end{align*}
\]
Theorem

Let $\mathcal{G}(V, E)$ be a chordal graph obtained from the polynomial $p(x)$ with maximal cliques $\{C_1, C - 2, ..., C_t\}$.

Then, polynomial $p(x)$ is SOS if and only if:

$p(x) \in SOS \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS$

$X_k$: Nodes in clique $C_k$

- Constraint of the form $p(x) \in SOS$ can be replaced by SOS constraints on low dimensional polynomials.


108

Fall 2019
Theorem
Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a chordal graph obtained from the polynomial $p(x)$ with maximal cliques $\{C_1, C - 2, ..., C_t\}$.

Then, polynomial $p(x)$ is SOS if and only if:

$$p(x) \in SOS \iff p(x) = \sum_k p_k(X_k) \quad \text{where} \quad p_k(X_k) \in SOS$$

$X_k$: Nodes in clique $C_k$

- Constraint of the form $p(x) \in SOS$ can be replaced by SOS constraints on **low dimensional** polynomials.

$$p(x_1, x_2, x_3) = 2(1 + x_1 + x_3 + x_1^2 + x_1 x_2 + x_2^2 + x_2 x_3 + x_3^2)$$

Coupled variables: $(x_1, x_2)$, $(x_2, x_3)$

Edges between coupled variables

Polynomial with sparsity pattern $\mathcal{G}(\mathcal{V}, \mathcal{E})$
Theorem

Let \( G(\mathcal{V}, \mathcal{E}) \) be a chordal graph obtained from the polynomial \( p(x) \) with maximal cliques \( \{C_1, C - 2, \ldots, C_i\} \).

Then, polynomial \( p(x) \) is SOS if and only if:

\[
p(x) \in \text{SOS} \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in \text{SOS}
\]

\( X_k \): Nodes in clique \( C_k \)

- Constraint of the form \( p(x) \in \text{SOS} \) can be replaced by SOS constraints on low dimensional polynomials.

\[
p(x_1, x_2, x_3) = 2(1 + x_1 + x_3 + x_1^2 + x_1 x_2 + x_2^2 + x_2 x_3 + x_3^2)
\]

Polynomial with sparsity pattern

Coupled variables: \((x_1, x_2), (x_2, x_3)\)

Edges between coupled variables

Maximal clique \( C_1 \)

Maximal clique \( C_2 \)
Theorem

Let \( G(V, E) \) be a chordal graph obtained from the polynomial \( p(x) \) with maximal cliques \( \{C_1, C_2, ..., C_t \} \)

Then, polynomial \( p(x) \) is SOS if and only if:

\[
p(x) \in \text{SOS} \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in \text{SOS}
\]

\( X_k \): Nodes in clique \( C_k \)

➢ Constraint of the form \( p(x) \in \text{SOS} \) can be replaced by SOS constraints on low dimensional polynomials.

\[
p(x_1, x_2, x_3) = 2(1 + x_1 + x_3 + x_1^2 + x_1x_2 + x_2^2 + x_2x_3 + x_3^2)
\]

Coupled variables: \( (x_1, x_2) \), \( (x_2, x_3) \)

Edges between coupled variables

\[
p(x_1, x_2, x_3) \in \text{SOS} \iff p(x_1, x_2) = p_1(x_1, x_2) \in \text{SOS} + p_2(x_2, x_3) \in \text{SOS}
\]

\[
p(x_1, x_2, x_3) = (1 + x_1)^2 + (x_1 + x_2)^2 + (1 + x_3^2)^2 + (x_2 + x_3)^2
\]
Theorem

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a chordal graph obtained from the polynomial $p(x)$ with maximal cliques $\{C_1, C-2, ..., C_l\}$.

Then, polynomial $p(x)$ is SOS if and only if:

$$p(x) \in SOS \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS$$

$X_k$: Nodes in clique $C_k$

- Constraint of the form $p(x) \in SOS$ can be replaced by SOS constraints on low dimensional polynomials.

$$p(x_1, x_2, x_3) = 2(1 + x_1 + x_3 + x_1^2 + x_1x_2 + x_2^2 + x_2x_3 + x_3^2)$$

Coupled variables: $(x_1, x_2), (x_2, x_3)$

Edges between coupled variables

$$p(x_1, x_2, x_3) \in SOS \iff p(x_1, x_2) = p_1(x_1, x_2) \in SOS + p_2(x_2, x_3) \in SOS$$

$$p(x_1, x_2, x_3) = (1 + x_1)^2 + (x_1 + x_2)^2 + (1 + x_3)^2 + (x_2 + x_3)^2$$

$$p(x_1, x_2, x_3) \in SOS \implies p(x_1, x_2, x_3) \in SSOS$$
Unconstrained optimization

\[
\begin{aligned}
m\min_x & \quad p(x) \\
\end{aligned}
\]

**SOS Program:**

\[
\begin{aligned}
\max_{Q \in S^n, \gamma} & \quad \gamma \\
\text{subject to} & \quad p(x) - \gamma \in SOS
\end{aligned}
\]

**SSOS Program:**

\[
\begin{aligned}
\max_{Q \in S^n, \gamma} & \quad \gamma \\
\text{subject to} & \quad p(x) - \gamma \in SSOS
\end{aligned}
\]
Unconstrained optimization

\[ \text{minimize } \quad p(x) \]

\underline{SOS Program:}

\[ \text{maximize } \quad \gamma \]

\[ \text{subject to } \quad p(x) - \gamma \in \text{SOS} \]

\underline{SSOS Program:}

\[ \text{maximize } \quad \gamma \]

\[ \text{subject to } \quad p(x) - \gamma \in \text{SSOS} \]

Constrained optimization

\[ \text{minimize } \quad \begin{array}{c} x \end{array} \quad p(x) \]

\[ \text{subject to } \quad g_i(x) \geq 0, \quad i = 1, \ldots, n \]

\underline{SOS Program:}

\[ \text{maximize } \quad \gamma, \sigma_i \]

\[ \text{subject to } \quad p(x) - \gamma - \sum_{i=1}^{m} \sigma_i(x) g_i(x) \in \text{SOS} \]

\[ \sigma_i(x) \in \text{SOS}_{2d_i}, \quad i = 0, \ldots, m \]

\underline{SSOS Program:}

\[ \text{maximize } \quad \gamma, \sigma_i \]

\[ \text{subject to } \quad p(x) - \gamma - \sum_{i=1}^{m} \sigma_i(x) g_i(x) \in \text{SSOS} \]

\[ \sigma_i(x) \in \text{SOS}_{2d_i}, \quad i = 0, \ldots, m \]

should preserve the correlative sparsity of \( g_i \)
\[ p(x) - \gamma - \sum_{i=1}^{m} \left( \sigma_i(x) g_i(x) \right) \in SSOS \]
\[ \sigma_i(x) \in SOS_{2d_i}, \; i = 0, \ldots, m \]

- \( \sigma_i(x) \) should preserve the correlative sparsity of \( g_i(x) \)

- **Example:**
  \[ g_i(\tilde{x}) \]: is a polynomial in terms of subset of variables \( \tilde{x} \)
  \[ \sigma_i(\tilde{x}) \]: SOS polynomial in terms of variables \( \tilde{x} \)

More information:


Example: [https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/Sparse_SOS/Example_SSOS_compare_Cons.m](https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/Sparse_SOS/Example_SSOS_compare_Cons.m)
Sparse SOS using Yalmip

1) Copy “corrsparsity.m” to the folder of /modules/sos, and replace the original corrsparsity.m.

https://github.com/zhengy09/sos_csp

2) Add the “ops.sos.csp = 1” to the Yalmip SOS optimization code.


sparsePOP 3.03 (MATLAB Package)

This package also provides the optimal solution $x^*$ of SSOS optimization.

https://sourceforge.net/projects/sparsepop/

Example 1: Unconstrained Optimization

\[ f_{cs}(x) = \sum_{i \in J} \left( (x_i + 10x_{i+1})^2 + 5(x_{i+2} - x_{i+3})^2 \right) + (x_{i+1} - 2x_{i+2})^4 + 10(x_i - 10x_{i+3})^4, \]

\[ J = \{1, 3, 5, \ldots, n - 3\} \]

<table>
<thead>
<tr>
<th>Number of variables</th>
<th>cl.str</th>
<th>( \epsilon_{obj} )</th>
<th>sparse</th>
<th>dense</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>3*14</td>
<td>3.5e-7</td>
<td>0.6</td>
<td>3059.5</td>
</tr>
<tr>
<td>40</td>
<td>3*38</td>
<td>8.4e-7</td>
<td>1.4</td>
<td>—</td>
</tr>
<tr>
<td>100</td>
<td>3*98</td>
<td>5.5e-7</td>
<td>3.8</td>
<td>—</td>
</tr>
<tr>
<td>200</td>
<td>3*198</td>
<td>3.0e-7</td>
<td>8.4</td>
<td>—</td>
</tr>
<tr>
<td>400</td>
<td>3*398</td>
<td>3.6e-7</td>
<td>19.3</td>
<td>—</td>
</tr>
</tbody>
</table>

Example 2: Unconstrained Optimization

\[ f_{bb}(x) = \sum_{i=1}^{n} \left( x_i (2 + 5x_i^2) + 1 - \sum_{j \in J_i} (1 + x_j) x_j \right)^2, \]

\[ J_i = \{ j \mid j \neq i, \max(1, i - 5) \leq j \leq \min(n, i + 1) \}. \]

<table>
<thead>
<tr>
<th>Broyden banded function</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n )</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
</tbody>
</table>

Illustrative Example:

\[
\begin{align*}
P^* &= \min_{x \in \mathbb{R}^n} \quad 5 + \sum_{i=1}^{n} (x_i - 1)^2 \\
p^* &= 5, \quad x^* = [1, 1, \ldots, 1]^T \in \mathbb{R}^n
\end{align*}
\]

<table>
<thead>
<tr>
<th>Method</th>
<th>Variables</th>
<th>Relaxation Order</th>
<th>Time (s)</th>
<th>(p^*)</th>
<th>(x^*)</th>
<th>SDP Solver</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOS</td>
<td>200</td>
<td>1</td>
<td>286.5458</td>
<td>5</td>
<td>([1, \ldots, 1]^T)</td>
<td>mosek</td>
</tr>
<tr>
<td>SDSOS</td>
<td>200</td>
<td>1</td>
<td>3.6338</td>
<td>5</td>
<td></td>
<td>mosek</td>
</tr>
<tr>
<td>DSOS</td>
<td>200</td>
<td>1</td>
<td>2.6824</td>
<td>5</td>
<td></td>
<td>mosek</td>
</tr>
<tr>
<td>Spars SOS</td>
<td>200</td>
<td>1</td>
<td>0.2374</td>
<td>5</td>
<td></td>
<td>mosek</td>
</tr>
<tr>
<td>Spars POP</td>
<td>200</td>
<td>1</td>
<td>0.95</td>
<td>5</td>
<td></td>
<td>sdpt3</td>
</tr>
</tbody>
</table>

https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/SDSOS-DSOS/Example_SDSOS_compare_Uncons.m

https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/Sparse_SOS/Example_SSOS_compare_Uncons.m
Topics

1) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS)
   Modified SOS optimization that results in LP and Second order cone program

2) Bounded degree SOS (BSOS)
   Modified SOS optimization that results in smaller SDP’s.

3) Spars Sum-of-Squares Optimization (SSOS)
   Takes advantage of sparsity of the original problem to generate smaller SDP.

4) Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)
   Combination of 2 and 3
Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)

- **Combines** Bounded degree SOS (BSOS) and Chordal-Sparse SOS.

- Takes advantages of sparsity of the original problem to reduce the size of the bounded degree SOS.

- **It relies on “Running Intersection Property” (Chordal sparsity of the graph)**

- **Example:** [https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/Sparse_Bounded_Degree_SOS/SBSOS_Example1.m](https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/Sparse_Bounded_Degree_SOS/SBSOS_Example1.m)

- **MATLAB Code**
  - [https://github.com/tweisser/Sparse_BSOS](https://github.com/tweisser/Sparse_BSOS)

  This package also provides the optimal solution $x^*$ of SBSOS optimization.
Example 1: Constrained Optimization (Chained Singular Function)

\[ f := \sum_{j \in H} \left( (x_j + 10x_{j+1})^2 + 5(x_{j+2} - x_{j+3})^2 + (x_{j+1} - 2x_{j+2})^4 + 10(x_j - x_{j+3})^4 \right) \]

\[ H := \{ 2i - 1 : i = 1, \ldots, n/2 - 1 \} \]

\[ K = \left\{ x \in \mathbb{R}^n : 1 - \sum_{i \in I_\ell} x_i \geq 0, \quad \ell = 1, \ldots, p; \quad x_i \geq 0, \quad i = 1, \ldots, n \right\} , \]


Application


<table>
<thead>
<tr>
<th>Number of variables</th>
<th>Chained Singular</th>
<th>rel.</th>
<th>Sparse-BSOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n = 500 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( d = 1 )</td>
<td></td>
<td>(-1.4485e-02)</td>
</tr>
<tr>
<td></td>
<td>( d = 2 )</td>
<td></td>
<td>(-9.7833e-10)</td>
</tr>
<tr>
<td>( n = 600 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( d = 1 )</td>
<td></td>
<td>(-2.7372e-03)</td>
</tr>
<tr>
<td></td>
<td>( d = 2 )</td>
<td></td>
<td>(-1.2640e-09)</td>
</tr>
<tr>
<td>( n = 700 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( d = 1 )</td>
<td></td>
<td>(-1.7548e-03)</td>
</tr>
<tr>
<td></td>
<td>( d = 2 )</td>
<td></td>
<td>(-1.7613e-09)</td>
</tr>
<tr>
<td>( n = 800 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( d = 1 )</td>
<td></td>
<td>(-1.9438e-03)</td>
</tr>
<tr>
<td></td>
<td>( d = 2 )</td>
<td></td>
<td>(2.1935e-09)</td>
</tr>
<tr>
<td>( n = 900 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( d = 1 )</td>
<td></td>
<td>(-1.8924e-02)</td>
</tr>
<tr>
<td></td>
<td>( d = 2 )</td>
<td></td>
<td>(-2.6072e-09)</td>
</tr>
<tr>
<td>( n = 1000 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( d = 1 )</td>
<td></td>
<td>(-4.4914e-02)</td>
</tr>
<tr>
<td></td>
<td>( d = 2 )</td>
<td></td>
<td>(-9.3508e-10)</td>
</tr>
</tbody>
</table>

Table 6  Comparison Sparse-BSOS (k = 2)

Solution | \( \text{rk} \) | Time (s) |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(-1.4485e-02)</td>
<td>1.0</td>
<td>19.6</td>
</tr>
<tr>
<td>(-9.7833e-10)</td>
<td>1</td>
<td>17.8</td>
</tr>
<tr>
<td>(-2.7372e-03)</td>
<td>1.0</td>
<td>40.1</td>
</tr>
<tr>
<td>(-1.2640e-09)</td>
<td>1</td>
<td>21.4</td>
</tr>
<tr>
<td>(-1.7548e-03)</td>
<td>1.0</td>
<td>41.6</td>
</tr>
<tr>
<td>(-1.7613e-09)</td>
<td>1</td>
<td>25.3</td>
</tr>
<tr>
<td>(-1.9438e-03)</td>
<td>1.0</td>
<td>58.9</td>
</tr>
<tr>
<td>(2.1935e-09)</td>
<td>1</td>
<td>29.0</td>
</tr>
<tr>
<td>(-1.8924e-02)</td>
<td>1.0</td>
<td>43.5</td>
</tr>
<tr>
<td>(-2.6072e-09)</td>
<td>1</td>
<td>33.5</td>
</tr>
<tr>
<td>(-4.4914e-02)</td>
<td>1.0</td>
<td>35.5</td>
</tr>
<tr>
<td>(-9.3508e-10)</td>
<td>1</td>
<td>39.5</td>
</tr>
</tbody>
</table>
1) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS)
   Modified SOS optimization that results in LP and Second order cone program

2) Bounded degree SOS (BSOS)
   Modified SOS optimization that results in smaller SDP’s.

3) Spars Sum-of-Squares Optimization (SSOS)
   Takes advantage of sparsity of the original problem to generate smaller SDP.

4) Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)
   Combination of 2 and 3
**Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)**

  
  Code: [https://github.com/tweisser/Sparse_BSOS](https://github.com/tweisser/Sparse_BSOS)

---

**Bounded Degree Sum-of-Squares Optimization (BSOS)**

  
  Code: [https://github.com/tweisser/Sparse_BSOS](https://github.com/tweisser/Sparse_BSOS)

---

**Sparse Sum-of-Squares Optimization (SSOS)**

  
  Code: [https://sourceforge.net/projects/sparsepop/](https://sourceforge.net/projects/sparsepop/)

  
  Code: [https://github.com/zhengy09/sos_csp](https://github.com/zhengy09/sos_csp)

---

**Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)**

  
  Code: [https://github.com/tweisser/Sparse_BSOS](https://github.com/tweisser/Sparse_BSOS)
Appendix I: SDSOS/DSOS Polynomials
### Sum-Of-Squares Polynomials

\[ p(x) \in \text{SOS} \quad \rightarrow \quad p(x) = \sum_{i=1}^{\ell} h_i^2(x) \]

where \( h_i(x) \in \mathbb{R}[x], \; i = 1, \ldots, \ell \)

\[ p(x) = B(x)^T Q B(x) \quad \text{where } Q \in S^n_+ \]

### Diagonally-Dominant-Sum-Of-Squares Polynomials

\[ p(x) \in \text{DSOS} \]

\[ p(x) = \sum_i \alpha_i m_i^2(x) + \sum_{i,j} \beta_{ij}^+(m_i(x) + m_j(x))^2 + \sum_{i,j} \beta_{ij}^-(m_i(x) - m_j(x))^2 \]

for some nonnegative scalars \( \alpha_i, \beta_{ij}^+, \beta_{ij}^- \)

\[ p(x) = B(x)^T Q B(x) \quad \text{where } Q \in S^n_{dd} \]

### Scaled-Diagonally-Dominant-Sum-Of-Squares Polynomials

\[ p(x) \in \text{SDSOS} \]

\[ p(x) = \sum_i \alpha_i m_i^2(x) + \sum_{i,j} (\tilde{\beta}_{ij}^+ m_i(x) + \tilde{\beta}_{ij}^- m_j(x))^2 + \sum_{i,j} (\tilde{\beta}_{ij}^- m_i(x) - \tilde{\beta}_{ij}^+ m_j(x))^2 \]

for some scalars \( \alpha_i \geq 0, \tilde{\beta}_{ij}^+, \tilde{\beta}_{ij}^- \)

\[ p(x) = B(x)^T Q B(x) \quad \text{where } Q \in S^n_{sdd} \]

\[ \text{DSOS} \subseteq \text{SDSOS} \subseteq \text{SOS} \quad S^n_{dd} \subseteq S^n_{sdd} \subseteq S^n_+ \]

---


Fall 2019
Appendix II: Convergence of LP Relaxation
\[ \begin{align*}
\text{P}^* = & \min_{x \in \mathbb{R}^n} \quad p(x) \\
\text{subject to} \quad & x \in \mathbf{K} = \{ x \in \mathbb{R}^n : g_i(x) \geq 0, \ i = 1, \ldots, m \}
\end{align*} \]

\[ \begin{align*}
\text{P}^* = & \max_{\gamma \in \mathbb{R}} \quad \gamma \\
\text{subject to} \quad & p(x) - \gamma \geq 0, \ \forall x \in \mathbf{K} \quad \text{optimal solution} \quad \gamma^* = p(x^*)
\end{align*} \]

**SDP Relaxation**

\[ p(x) - \gamma^* = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x)g_i(x) \quad \exists \ \gamma^* \in \mathbb{R}, \ \sigma_0(x) \in SOS_{2d}, \ \sigma_i(x) \in SOS_{2d_i}, i = 1, \ldots, m \]

If \( \gamma^* = p(x^*) = P^* \quad p(x^*) - \gamma^* = 0 \quad \sigma_0(x^*) + \sum_{i=1}^{m} \sigma_i(x^*)g_i(x^*) = 0 \]
\( P^* = \min_{x \in \mathbb{R}^n} p(x) \)

subject to \( x \in K = \{ x \in \mathbb{R}^n : g_i(x) \geq 0, \ i = 1, \ldots, m \} \)

\( P^* = \max_{\gamma \in \mathbb{R}} \gamma \)

subject to \( p(x) - \gamma \geq 0, \ \forall x \in K \)

\( \gamma^* = p(x^*) \) \[ \text{optimal solution} \]

---

**SDP Relaxation**

\[ p(x) - \gamma^* = \sigma_0(x) + \sum_{i=1}^m \sigma_i(x)g_i(x) \]  
\[ \exists \gamma^* \in \mathbb{R}, \sigma_0(x) \in SOS_{2d}, \ \sigma_i(x) \in SOS_{2d}, i = 1, \ldots, m \]

if \( \gamma^* = p(x^*) = P^* \)  
\[ p(x^*) - \gamma^* = 0 \]

\[ \sigma_0(x^*) + \sum_{i=1}^m \sigma_i(x^*)g_i(x^*) = 0 \]

\( \text{if } x^* \in \text{int}K \)

\[ \sigma_0(x^*) + \sum_{i=1}^m \sigma_i(x^*)g_i(x^*) = 0 \]

\[ g_i(x^*) > 0 \]

\[ \text{Hence, This constraint is imposed by} \]

\[ \sigma_i(x^*) \ i = 0, \ldots, m \]

\( (\text{The same situation for } x^* \in \partial K) \)
\[
P^* = \min_x x^2 - 2x + 2
\]
subject to \( x \in K = \{ x : x(2-x) \geq 0 \} \)

\[
P_{sos}^* = \max_{\gamma \in \mathbb{R}, \sigma_0(x) \in SOS, \sigma_1(x) \in SOS} \gamma
\]
subject to \( x^2 - 2x + 2 - \gamma = \sigma_0(x) + \sigma_1(x) x(2-x) \)

\[
\gamma^* = 1 \quad \rightarrow \quad x^* = 1
\]

\[
\sigma_0(x) = (-0.291570596593 - 0.0571934472478x1 + 0.348740011438x1^2)^2 + (-0.956549252584 + 1.50888962843x1 - 0.552282590362x1^2)^2
\]

\[
\sigma_1(x) = (-0.653185546681 + 0.653173513801x1)^2
\]

\[\Rightarrow \text{At } x^* = 1 \in \text{int } K\]

\[
p(x^*) - \gamma^* = 0
\]

\[
\frac{\sigma_0(x^*) + \sigma_1(x^*)x^*(2-x^*)}{0} = 0
\]
\[ \text{LP Relaxation} \]

\[ p(x) - \gamma^* = \sum \lambda_{\alpha \beta}^* g_1^{\alpha_1}(x) \cdots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \cdots (1 - g_m(x))^{\beta_m} \]

\[ \text{if } \gamma^* = p(x^*) = P^* \quad p(x^*) - \gamma^* = 0 \quad \sum \lambda_{\alpha \beta}^* g_1^{\alpha_1}(x^*) \cdots g_m^{\alpha_m}(x^*)(1 - g_1(x^*))^{\beta_1} \cdots (1 - g_m(x^*))^{\beta_m} = 0 \]

\[ \exists \gamma^* \in \mathbb{R}, \lambda_{\alpha \beta}^* \geq 0 \]

\[ \text{Equivalent problems} \]

\[ P^* = \min_{x \in \mathbb{R}^n} p(x) \]

\[ \text{subject to} \quad x \in K = \{ x \in \mathbb{R}^n : 0 \leq g_i(x) \leq 1, \ i = 1, \ldots, m \} \]

\[ P^* = \max_{\gamma \in \mathbb{R}} \gamma \]

\[ \text{subject to} \quad p(x) - \gamma \geq 0, \ \forall x \in K \]

\[ \text{optimal solution} \quad \gamma^* = p(x^*) \]

\[ \gamma^* = p(x^*) \]

\[ \gamma^* \in \mathbb{R} \]

\[ x \]

\[ p(x) \]

\[ \gamma \in \mathbb{R} \]

\[ K \]

---

* Section 5.4.2, Jean Bernard Lasserre, “Moments, Positive Polynomials and Their Applications” Imperial College Press Optimization Series, V. 1, 2009.
\[ P^* = \min_{x \in \mathbb{R}^n} \quad p(x) \]
\[ \text{subject to} \quad x \in K = \{ x \in \mathbb{R}^n : 0 \leq g_i(x) \leq 1, \ i = 1, \ldots, m \} \]

\[ P^* = \max_{\gamma \in \mathbb{R}} \quad \gamma \]
\[ \text{subject to} \quad p(x) - \gamma \geq 0, \ \forall x \in K \]

**LP Relaxation**

\[ p(x) - \gamma^* = \sum \lambda_{\alpha, \beta}^* g_1^{\alpha_1}(x) \ldots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \ldots (1 - g_m(x))^{\beta_m} \]

\[ \exists \ \gamma^* \in \mathbb{R}, \ \lambda_{\alpha, \beta}^* \geq 0 \]

\[ \text{if } \gamma^* = p(x^*) = P^* \quad p(x^*) - \gamma^* = 0 \quad \sum \lambda_{\alpha, \beta}^* g_1^{\alpha_1}(x^*) \ldots g_m^{\alpha_m}(x^*)(1 - g_1(x^*))^{\beta_1} \ldots (1 - g_m(x^*))^{\beta_m} = 0 \]

\[ \text{if } x^* \in \text{int} K \quad \sum \lambda_{\alpha, \beta}^* g_1^{\alpha_1}(x^*) \ldots g_m^{\alpha_m}(x^*)(1 - g_1(x^*))^{\beta_1} \ldots (1 - g_m(x^*))^{\beta_m} > 0 \]

\[ g_i(x^*) > 0 \quad 1 - g_i(x^*) > 0 \]

---

Section 5.4.2, Jean Bernard Lasserre, “Moments, Positive Polynomials and Their Applications” Imperial College Press Optimization Series, V. 1, 2009.
\[ \begin{align*}
\text{P}^* & = \min_{x \in \mathbb{R}^n} \quad p(x) \\
\text{subject to} & \quad x \in K = \{ x \in \mathbb{R}^n : 0 \leq g_i(x) \leq 1, \; i = 1, \ldots, m \} \\
\text{P}^* & = \max_{\gamma \in \mathbb{R}} \quad \gamma \\
\text{subject to} & \quad p(x) - \gamma \geq 0, \; \forall x \in K \\
\end{align*} \]

Equivalent problems

\[ \gamma^* = p(x^*) \]

\[ \gamma \in \mathbb{R} \]

LP Relaxation

\[ p(x) - \gamma^* = \sum \lambda_{\alpha \beta}^* g_1^{\alpha_1}(x) \cdots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \cdots (1 - g_m(x))^{\beta_m} \]

\[ \exists \gamma^* \in \mathbb{R}, \lambda_{\alpha \beta}^* \geq 0 \]

if \( \gamma^* = p(x^*) = P^* \)

\[ p(x^*) - \gamma^* = 0 \]

\[ \sum \lambda_{\alpha \beta}^* g_1^{\alpha_1}(x^*) \cdots g_m^{\alpha_m}(x^*)(1 - g_1(x^*))^{\beta_1} \cdots (1 - g_m(x^*))^{\beta_m} = 0 \]

if \( x^* \in \text{int}K \)

\[ \sum \lambda_{\alpha \beta}^* g_1^{\alpha_1}(x^*) \cdots g_m^{\alpha_m}(x^*)(1 - g_1(x^*))^{\beta_1} \cdots (1 - g_m(x^*))^{\beta_m} > 0 \]

\[ \sum_{j=1}^m \alpha_j + \beta_j \leq d \]

Convergences to zero

\[ \lim_{d \to \infty} P_{L^d} = P^* \]

\[ \gamma^* (\text{optimal solution of the original problem}) \text{ cannot be attained.} \]

\[ \gamma^* = p(x^*) \]

Section 5.4.2, Jean Bernard Lasserre, “Moments, Positive Polynomials and Their Applications” Imperial College Press Optimization Series, V. 1, 2009.
Example:

\[
P^* = \min_{x \in \mathbb{R}^n} \quad p(x) = x^2 - x
\]

subject to \( x \in K = \{ x \in \mathbb{R}^n : g_1(x) = x \geq 0, \ g_2(x) = 1 - x \geq 0 \} \)

\[
x^* = \frac{1}{2} \in \text{int}K
\]
\[
p(x^*) = -0.25
\]

LP Relaxation

\[
P_L^* = \max_{\gamma, \lambda \alpha \beta \geq 0} \quad \gamma
\]

subject to \( p(x) - \gamma = \sum \lambda_{\alpha \beta} g_1^{\alpha_1}(x) \ldots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \ldots (1 - g_m(x))^{\beta_m} \)

\( \forall \alpha, \beta \in \mathbb{N}^n \)
\[\sum_{j=1}^{m} \alpha_j + \beta_j \leq i \]

Slow monotone convergence to \(-0.25\):
\[
P_L^2 = -\frac{1}{3} \quad P_L^4 = -\frac{1}{3} \quad P_L^6 = -0.3 \quad P_L^{10} = -0.27 \quad P_L^{15} = -0.2695
\]

Example:

\[
P^* = \min_{x \in \mathbb{R}^n} \quad p(x) = x - x^2
\]

subject to \( x \in K = \{ x \in \mathbb{R}^n : g_1(x) = x \geq 0, \ g_2(x) = 1 - x \geq 0 \} \)

\[
x^* = 0.1 \in \partial K
\]
\[
p(x^*) = 0
\]

LP Representation

\[
p(x) - \gamma^* = g_1(x)g_2(x)
\]
\[
x - x^2 = x(1 - x)
\]

Some of \( g_i(x)'s, (1 - g_i(x))'s \) are zero. Hence, finite convergence can take place.

Appendix III: Bounded Degree SOS
Lagrangian Perspective
To gain more insight into how the BSOS optimization works, consider the following Nonlinear optimization and its dual:

\[
P^* = \min_{x \in \mathbb{R}^n} \quad p(x)
\]
subject to \( g_1^{\alpha_1}(x) \cdots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \cdots (1 - g_m(x))^{\beta_m} \geq 0, \quad \forall \sum_{j=1}^m \alpha_j + \beta_j \leq d \)

**Lagrange function**

\[
L(\lambda, x) = p(x) - \sum_{j=1}^m \lambda_1^{\alpha_1} g_1^{\alpha_1}(x) \cdots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \cdots (1 - g_m(x))^{\beta_m}
\]
subject to \( \sum_{j=1}^m \alpha_j + \beta_j \leq d \)

**Dual Optimization:**

\[
P_{dual}^* = \max_{\lambda} \min_{x \in \mathbb{R}^n} \quad L(x, \lambda)
\]
subject to \( \lambda \geq 0 \)

To solve \( \min_{x \in \mathbb{R}^n} L(x, \lambda) \), we can use SOS relaxation.

\[
\max_{\gamma} \gamma
\]
subject to \( L(x, \lambda) - \gamma \geq 0 \)

\[
\max_{\gamma, Q_0 \geq 0} \gamma
\]
subject to \( L(x, \lambda) - \gamma \in SOS_k \)

This results in BSOS formulation.
\[
\begin{align*}
\text{maximize} \quad & \gamma \\
\text{subject to} \quad & L(x, \lambda) - \gamma \in SOS_k
\end{align*}
\]

- For \( k = 0 \), this is results in “Krivine-Stengle’s Positivity Certificate” based LP. (brutal simplification of \( \min_{x \in \mathbb{R}^n} L(x, \lambda) \))

- For \( k > 0 \), this is results in “BSOS” relaxation. (tractable simplification of \( \min_{x \in \mathbb{R}^n} L(x, \lambda) \))
\[
\begin{align*}
\max_{\gamma, Q_0 \geq 0} & \quad \gamma \\
\text{subject to} & \quad L(x, \lambda) - \gamma \in \text{SOS}_k
\end{align*}
\]

- For \( k = 0 \), this is results in “Krivine-Stengle’s Positivity Certificate” based LP. (brutal simplification of \( \min_{x \in \mathbb{R}^n} L(x, \lambda) \))

- For \( k > 0 \), this is results in “BSOS” relaxation. (tractable simplification of \( \min_{x \in \mathbb{R}^n} L(x, \lambda) \))

\[\Rightarrow \text{Hence, } \lambda_{\alpha \beta} \text{ in LP and BSOS are approximation of the Lagrange multipliers.}\]

\[\Rightarrow \text{Based on KKT optimality condition: } \lambda_{\alpha \beta} g_1^{\alpha_1}(x^*) \ldots g_m^{\alpha_m}(x^*) (1 - g_1(x^*))^{\beta_1} \ldots (1 - g_m(x^*))^{\beta_m} = 0\]

\[\Rightarrow \text{Hence, when finite convergence in BSOS occurs: } \lambda_{\alpha \beta} g_1^{\alpha_1}(x^*) \ldots g_m^{\alpha_m}(x^*) (1 - g_1(x^*))^{\beta_1} \ldots (1 - g_m(x^*))^{\beta_m} = 0 \]

\[\Rightarrow p(x^*) - \gamma^* = 0\]
\[
\begin{align*}
\text{maximize} \quad & \gamma \\
\text{subject to} \quad & L(x, \lambda) - \gamma \in \text{SOS}_k
\end{align*}
\]

- For \( k = 0 \), this results in “Krivine-Stengle’s Positivity Certificate” based LP.
  (brutal simplification of \( \min_{x \in \mathbb{R}^n} L(x, \lambda) \))
- For \( k > 0 \), this results in “BSOS” relaxation.
  (tractable simplification of \( \min_{x \in \mathbb{R}^n} L(x, \lambda) \))

\[\begin{align*}
\text{Hence,} \quad & \lambda_{\alpha\beta} \quad \text{in LP and BSOS are approximation of the Lagrange multipliers.} \\
\text{Based on KKT optimality condition:} \quad & \lambda_{\alpha\beta} g_1^{\alpha_1}(x^*) \cdots g_m^{\alpha_m}(x^*)(1 - g_1(x^*))^{\beta_1} \cdots (1 - g_m(x^*))^{\beta_m} = 0 \\
\text{Hence, when finite convergence in BSOS occurs:} \quad & \lambda_{\alpha\beta} g_1^{\alpha_1}(x^*) \cdots g_m^{\alpha_m}(x^*)(1 - g_1(x^*))^{\beta_1} \cdots (1 - g_m(x^*))^{\beta_m} = 0
\end{align*}\]

Appendix IV: Maximal Clique and Principal Submatrix
Maximal Clique and Principal Submatrix

- Matrix $X \in \mathcal{S}^n$ with sparsity pattern defined by Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

- $\mathcal{C}_k$ is maximal clique of graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with $|\mathcal{C}_k|$ nodes.

- Define matrix $E_{\mathcal{C}_k} \in \mathbb{R}^{|\mathcal{C}_k| \times n}$ as follows:

$$[E_{\mathcal{C}_k}]_{i,j} = \begin{cases} 1, & \text{if } \mathcal{C}_k(i) = j \\ 0, & \text{otherwise} \end{cases}$$

Where $\mathcal{C}_k(i)$ is $i$-th node in $\mathcal{C}_k$.

$E_{\mathcal{C}_1} \in \mathbb{R}^{2 \times 3}$, $E_{\mathcal{C}_1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ nodes in $\mathcal{C}_1$.

$E_{\mathcal{C}_2} \in \mathbb{R}^{2 \times 3}$, $E_{\mathcal{C}_2} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ nodes in $\mathcal{C}_2$.

$X_{\mathcal{C}_1} = E_{\mathcal{C}_1}X_{\mathcal{C}_1}^T = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}$.

$X_{\mathcal{C}_2} = E_{\mathcal{C}_2}X_{\mathcal{C}_2}^T = \begin{bmatrix} 0.5 & 1 \\ 1 & 2 \end{bmatrix}$.

$X = E_{\mathcal{C}_1}^TX_{\mathcal{C}_1}E_{\mathcal{C}_1} + E_{\mathcal{C}_2}^TX_{\mathcal{C}_2}E_{\mathcal{C}_2} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$.