SparseJSR: A Fast Algorithm to Compute Joint Spectral Radius via Sparse SOS Decompositions

Jie Wang
Joint work with Martina Maggio and Victor Magron

LAAS-CNRS

ACC 2021

Contents

1. Computing JSR via sum-of-squares (SOS) programming
2. Term sparsity in SOS decompositions
3. Computing JSR via a hierarchy of sparse SOS programs
4. Numerical experiments

Joint spectral radius

Given $\mathcal{A} = \{A_1, \ldots, A_m\} \subseteq \mathbb{R}^{n\times n}$, the joint spectral radius (JSR) of $\mathcal{A}$ is defined by

$$\rho(\mathcal{A}) := \lim_{k\to\infty} \max_{\sigma\in\{1,\ldots,m\}^k} ||A_{\sigma_1} A_{\sigma_2} \cdots A_{\sigma_k}||^{\frac{1}{k}}.$$

Joint spectral radius

Given $\mathcal{A} = \{A_1, \ldots, A_m\} \subseteq \mathbb{R}^{n \times n}$, the joint spectral radius (JSR) of $\mathcal{A}$ is defined by

$$\rho(\mathcal{A}) := \lim_{k\to\infty} \max_{\sigma\in\{1,\ldots,m\}^k} ||A_{\sigma_1} A_{\sigma_2} \cdots A_{\sigma_k}||^{\frac{1}{k}} .$$

Applications:

- stability of switched linear dynamical systems
- continuity of wavelet functions
- combinatorics and language theory
- capacity of some codes
- trackability of graphs

Bounding JSR from below and from above

▷ NP-hard to compute/approximate

A lot of algorithms aiming to compute lower/upper bounds for JSR:

- Gripenberg's algorithm; (lower bound)
- Balanced polytope method; (upper bound)
- Lifted polytope method; (upper bound)
- The ellipsoid method; (upper bound)
- SOS programming; (upper bound)

Typically, it is more difficult to give a good upper bound than to give a good lower bound.

Positive forms and JSR

Theorem (Parrilo and Jadbabaie, 2008)

Given a set of matrices $\mathcal{A} = \{A_1, \ldots, A_m\} \subseteq \mathbb{R}^{n \times n}$*, let* p *be a strictly positive form of degree* $2d$ *that satisfies*

$$p(A_i \mathrm{x}) \leq \gamma^{2d} p(\mathrm{x}), \quad \forall \mathrm{x} \in \mathbb{R}^n, \quad i = 1, \ldots, m.$$

Then, $\rho(\mathcal{A}) \leq \gamma$.

Form: homogeneous polynomial

A SOS program for approximating JSR

- SOS forms ($\Sigma_{n,2d}$): $f = \sum_{i=1}^{t} f_i^2 \in \mathbb{R}[\mathrm{x}]_{2d}$

Inspired by the previous theorem and replacing positive forms by more tractable SOS forms, consider the SOS program (for every $d \geq 1$):

$$\rho_{\mathrm{SOS},2d}(\mathcal{A}) := \inf_{p \in \mathbb{R}[\mathrm{x}]_{2d}, \gamma} \gamma$$
$$\text{s.t.} \quad \begin{cases} p(\mathrm{x}) - ||\mathrm{x}||_2^{2d} \in \Sigma_{n,2d}, \\ \gamma^{2d} p(\mathrm{x}) - p(A_i \mathrm{x}) \in \Sigma_{n,2d}, \quad i = 1, \ldots, m. \end{cases}$$

This can be solved via semidefinite programming (SDP) by bisection on γ.

A SOS program for approximating JSR

• SOS forms ($\Sigma_{n,2d}$): $f = \sum_{i=1}^{t} f_i^2 \in \mathbb{R}[x]_{2d}$

Inspired by the previous theorem and replacing positive forms by more tractable SOS forms, consider the SOS program (for every $d \geq 1$):

$$\rho_{\mathrm{SOS},2d}(\mathcal{A}) := \inf_{p \in \mathbb{R}[x]_{2d}, \gamma} \gamma$$

$$\text{s.t.} \quad \begin{cases} p(x) - ||x||_2^{2d} \in \Sigma_{n,2d}, \\ \gamma^{2d} p(x) - p(A_i x) \in \Sigma_{n,2d}, \quad i = 1, \ldots, m. \end{cases}$$

This can be solved via semidefinite programming (SDP) by bisection on γ.

Theorem (Parrilo and Jadbabaie, 2008)

Let $\mathcal{A} = \{A_1, \ldots, A_m\} \subseteq \mathbb{R}^{n \times n}$. For any integer $d \geq 1$, it holds $m^{-\frac{1}{2d}} \rho_{SOS,2d}(\mathcal{A}) \leq \rho(\mathcal{A}) \leq \rho_{SOS,2d}(\mathcal{A})$.

Scalability issue

The size of SDP at relaxation order d:

- number of PSD matrices: $m+1$
- size of PSD matrices: $\binom{n-1+d}{d}$

In view of the current state of SDP solvers, the problem size is quite limited.

Scalability issue

The size of SDP at relaxation order d:

- number of PSD matrices: $m + 1$
- size of PSD matrices: $\binom{n-1+d}{d}$

In view of the current state of SDP solvers, the problem size is quite limited.

Motivation: Can we exploit the sparsity in $A_1, \ldots, A_m$ to improve the scalability of the SOS approach for computing JSR?

Checking SOS via SDP

- standard monomial basis $V_d(\mathrm{x}) = [x_1^d, \ldots, x_n^d]$: all monomials of degree d
- $f \in \Sigma_{n,2d} \iff \exists Q \succeq 0$ (Gram matrix) such that $f = V_d(\mathrm{x}) Q V_d(\mathrm{x})^T$
  $\rightsquigarrow$ SDP

Checking SOS via SDP

- standard monomial basis $V_d(\mathrm{x}) = [x_1^d, \ldots, x_n^d]$: all monomials of degree d
- $f \in \Sigma_{n,2d} \iff \exists Q \succeq 0$ (Gram matrix) such that $f = V_d(\mathrm{x}) Q V_d(\mathrm{x})^T$ $\rightsquigarrow$ SDP

When f is sparse:

$\triangleright$ Generate a smaller monomial basis $\mathscr{B} \subseteq V_d(\mathrm{x})$

$\triangleright$ Exploit the block structure of Q

Generating a smaller monomial basis

- The monomial basis given by the Newton polytope:

$$\mathscr{B} = \{\mathbf{x}^{\boldsymbol{\beta}} \mid \boldsymbol{\beta} \in \frac{1}{2}\mathrm{New}(f) \cap \mathbb{N}^n\}$$

Generating a smaller monomial basis

• The monomial basis given by the Newton polytope:

$$\mathscr{B} = \{\mathrm{x}^{\boldsymbol{\beta}} \mid \boldsymbol{\beta} \in \frac{1}{2}\mathrm{New}(f) \cap \mathbb{N}^n\}$$

• Alternatively, we may use a heuristic method which might generate a smaller monomial basis:

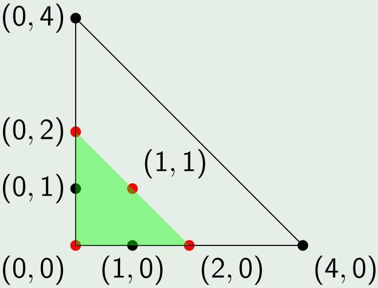
$$\mathscr{B} = \{\mathrm{x}^{\boldsymbol{\beta}} \in V_d(\mathrm{x}) \mid \exists \mathrm{x}^{\gamma} \in V_d(\mathrm{x}) \text{ s.t. } \mathrm{x}^{\boldsymbol{\beta}} \cdot \mathrm{x}^{\gamma} \in \mathrm{supp}(f)\}$$

(For $f = \sum_{\boldsymbol{\alpha}} f_{\boldsymbol{\alpha}} \mathrm{x}^{\boldsymbol{\alpha}} \in \mathbb{R}[\mathrm{x}]$, $\mathrm{supp}(f) := \{\mathrm{x}^{\boldsymbol{\alpha}} \mid f_{\boldsymbol{\alpha}} \neq 0\}$)

Generating a smaller monomial basis

Example

Let $f = 1 + x^4 + y^4 + x^2y^2$.



Chordal graphs

• S_G: the set of symmetric matrices with sparsity pattern represented by the graph G

• chordal graph: any cycle of length at least four has a chord

• chordal extension: a chordal graph $\overline{G}(V, \overline{E})$ containing $G(V, E)$ as a subgraph

$\triangleright$ approximately smallest chordal extension (aiming to minimize the clique number)

$\triangleright$ maximal chordal extension (completing each connected component)

Decompositions of PSD matrices

Given a maximal clique C of $G(V, E)$, define $P_C \in \mathbb{R}^{|C|\times|V|}$ as

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

where $C(i)$ denotes the i-th node in C, sorted in the ordering compatible with V.

Decompositions of PSD matrices

Given a maximal clique C of $G(V, E)$, define $P_C \in \mathbb{R}^{|C| \times |V|}$ as

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

where $C(i)$ denotes the i-th node in C, sorted in the ordering compatible with V.

Theorem (Agler, 1988)

Let $G(V, E)$ be a chordal graph and assume that $C_1, \ldots, C_t$ are the list of maximal cliques of $G(V, E)$. Then a matrix $Q \in \mathbf{S}_+^{|V|} \cap \mathbf{S}_G$ if and only if there exists $Q_k \in \mathbf{S}_+^{|C_k|}$ for $k = 1, \ldots, t$ such that $Q = \sum_{k=1}^{t} P_{C_k}^T Q_k P_{C_k}$.

Decompositions of PSD matrices

Example

$$\begin{bmatrix}1 & 1 & 0\\ 1 & 2 & 1\\ 0 & 1 & 1\end{bmatrix} (\succeq 0) = \begin{bmatrix}1 & 1 & 0\\ 1 & 1 & 0\\ 0 & 0 & 0\end{bmatrix} (\succeq 0) + \begin{bmatrix}0 & 0 & 0\\ 0 & 1 & 1\\ 0 & 1 & 1\end{bmatrix} (\succeq 0).$$

Term sparsity pattern

Term sparsity pattern graph $G(V, E)$

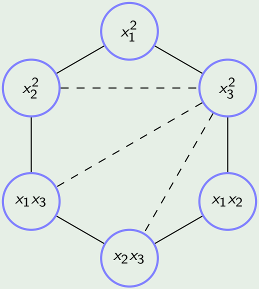
▷ $V := \mathscr{B}$ (the monomial basis)

▷ $\{\mathrm{x}^{\boldsymbol{\alpha}}, \mathrm{x}^{\boldsymbol{\beta}}\} \in E \iff \mathrm{x}^{\boldsymbol{\alpha}} \cdot \mathrm{x}^{\boldsymbol{\beta}} = \mathrm{x}^{\boldsymbol{\alpha}+\boldsymbol{\beta}} \in \mathrm{supp}(f)$

Example

Let $f = x_1^4 + x_2^4 + x_3^4 + x_1^2x_2^2 + x_1^2x_3^2 + x_1x_2x_3^2 + x_1x_2^2x_3$.

Figure: The term sparsity pattern graph of f and a chordal extension



Sparse SOS decompositions

- $\Sigma_{\mathscr{A}} := \{f \in \mathbb{R}[\mathscr{A}] \mid \exists Q \in \mathsf{S}_+^{|\mathscr{B}|} \cap \mathsf{S}_{\overline{G}} \text{ s.t. } f = \mathscr{B} Q \mathscr{B}^T\}$

Theorem (Wang, Li and Xia, 2019; Wang, Magron and Lasserre, 2021)

Given $\mathscr{A} \subseteq V_{2d}(\mathrm{x})$, assume that $\mathscr{B}$ is a monomial basis and G is the term sparsity pattern graph. Let $\mathscr{B}_1, \ldots, \mathscr{B}_t \subseteq V$ be the list of maximal cliques of $\overline{G}$ (a chordal extension of G). Then, $f \in \Sigma_{\mathscr{A}}$ if and only if there exists $f_k = \mathscr{B}_k Q_k \mathscr{B}_k^T$ with $Q_k \in \mathsf{S}_+^{|\mathscr{B}_k|}$ for $k = 1, \ldots, t$ such that $f = \sum_{k=1}^t f_k$.

Key Message: By the above theorem, checking membership in $\Sigma_{\mathscr{A}}$ boils down to solving an SDP involving PSD matrices of small sizes if each maximal clique of $\overline{G}$ has a small size relative to the original matrix.

Generating a hierarchy of sparse supports

In order to invoke sparse SOS decompositions, we need to construct a sparse support for p in coordination with the sparsity of $\mathcal{A}$.

Fixing $d \geq 1$, let $p_0(\mathrm{x}) = \sum_{j=1}^{n} c_j x_j^{2d}$ with generic coefficients and let $\mathscr{A}^{(0)} = \mathrm{supp}(p_0)$. Then for $s \in \mathbb{N} \backslash \{0\}$, we iteratively define

$$\mathscr{A}^{(s)} := \mathscr{A}^{(s-1)} \cup \bigcup_{i=1}^{m} \mathrm{supp}(p_{s-1}(A_i \mathrm{x})),$$

where $p_{s-1}(\mathrm{x}) = \sum_{\mathrm{x}^{\boldsymbol{\alpha}} \in \mathscr{A}^{(s-1)}} c_{\boldsymbol{\alpha}} \mathrm{x}^{\boldsymbol{\alpha}}$ with generic coefficients. Then,

$$\mathscr{A}^{(1)} \subseteq \cdots \subseteq \mathscr{A}^{(s)} \subseteq \mathscr{A}^{(s+1)} \subseteq \cdots \subseteq V_{2d}(\mathrm{x})$$

A hierarchy of sparse SOS programs for computing JSR

Fixing $d \geq 1$, we can now consider the following hierarchy of sparse SOS programs indexed by $s \geq 1$ (which is called the sparse order):

$$\rho_{s,2d}(\mathcal{A}) := \inf_{p \in \mathbb{R}[\mathscr{A}^{(s)}], \gamma} \gamma$$
$$\text{s.t.} \quad \begin{cases} p(\mathrm{x}) - ||\mathrm{x}||_2^{2d} \in \Sigma_{\mathscr{A}^{(s)}}, \\ \gamma^{2d} p(\mathrm{x}) - p(A_i \mathrm{x}) \in \Sigma_{\mathscr{A}_i^{(s)}}, \quad i = 1, \ldots, m, \end{cases}$$

where $\mathscr{A}_i^{(s)} = \mathscr{A}^{(s)} \cup \mathrm{supp}(p_s(A_i \mathrm{x}))$.

A hierarchy of sparse SOS programs for computing JSR

Fixing $d \geq 1$, we can now consider the following hierarchy of sparse SOS programs indexed by $s \geq 1$ (which is called the sparse order):

$$\rho_{s,2d}(\mathcal{A}) := \inf_{p \in \mathbb{R}[\mathscr{A}^{(s)}],\gamma} \gamma$$
$$\text{s.t.} \quad \begin{cases} p(\mathrm{x}) - ||\mathrm{x}||_2^{2d} \in \Sigma_{\mathscr{A}^{(s)}}, \\ \gamma^{2d} p(\mathrm{x}) - p(A_i\mathrm{x}) \in \Sigma_{\mathscr{A}_i^{(s)}}, \quad i = 1, \ldots, m, \end{cases}$$

where $\mathscr{A}_i^{(s)} = \mathscr{A}^{(s)} \cup \mathrm{supp}(p_s(A_i\mathrm{x}))$.

Theorem (Wang, Maggio and Magron, 2021)

Let $\mathcal{A} = \{A_1, \ldots, A_m\} \subseteq \mathbb{R}^{n \times n}$. For any integer $d \geq 1$, it holds $\rho_{SOS,2d}(\mathcal{A}) \leq \cdots \leq \rho_{s,2d}(\mathcal{A}) \leq \cdots \leq \rho_{2,2d}(\mathcal{A}) \leq \rho_{1,2d}(\mathcal{A})$.

The SparseJSR software

The sparse approach has been implemented in the Julia package SparseJSR.

- ▷ ChordalGraph: generate approximately smallest chordal extensions
- ▷ JuMP: construct SDP
- ▷ MOSEK: solve SDP

The SparseJSR software

The sparse approach has been implemented in the Julia package SparseJSR.

- ▷ ChordalGraph: generate approximately smallest chordal extensions
- ▷ JuMP: construct SDP
- ▷ MOSEK: solve SDP

Settings for numerical experiments

- Sparse order: $s = 1$
- Tolerance for bisection: $\epsilon = 1 \times 10^{-5}$
- Initial interval for bisection: $[0, 2]$
- Lower bound (lb): Gripenberg's algorithm

Randomly generated examples

Here test matrices are randomly generated with sparsity pattern represented by a directed graph with n nodes and $n+10$ edges.

Table: Randomly generated examples with $d=1$ and $m=2$, ub: upper bound, mb: maximal size of SDP blocks, -: running time $>$ 3600s

n	lb	Sparse ($d=1$)			Dense ($d=1$)		
		time (s)	ub	mb	time (s)	ub	mb
30	0.8502	1.65	0.8666	10	7.79	0.8523	30
40	0.9446	2.68	0.9446	14	25.6	0.9446	40
50	0.8838	2.97	0.9102	14	55.9	0.8838	50
60	0.7612	3.64	0.7843	13	171	0.7612	60
70	0.9629	4.35	0.9629	11	308	0.9629	70
80	0.9345	5.95	0.9399	15	743	0.9345	80
90	0.8020	6.27	0.8465	14	1282	0.8020	90
100	0.8642	8.15	0.9132	13	2568	0.8659	100
110	0.8355	9.59	0.8839	15	-	-	-
120	0.7483	11.7	0.7735	16	-	-	-

Examples from control systems (I)

Here test matrices are taken from certain switched linear dynamical systems related to deadline misses.

Table: Examples from control systems with $d = 1$ and $m = 5$, ub: upper bound, mb: maximal size of SDP blocks, -: running time > 3600s, $*$: out of memory

n	lb	Sparse ($d = 1$)			Dense ($d = 1$)		
		time (s)	ub	mb	time (s)	ub	mb
20	0.8142	1.62	0.8142	12	9.08	0.8142	20
30	1.0924	4.42	1.0961	14	65.4	1.0961	30
40	0.9772	9.69	0.9804	16	259	0.9804	30
50	1.1884	17.5	1.1884	18	680	1.1884	50
60	1.3259	30.7	1.3259	20	1776	1.3259	60
70	1.2727	53.9	1.2727	22	-	-	-
80	1.4262	85.6	1.4262	24	-	-	-
90	1.4452	132	1.4452	26	-	-	-
100	1.5267	195	1.5267	28	$*$	$*$	$*$
110	1.5753	287	1.5753	30	$*$	$*$	$*$

Examples from control systems (II)

Here test matrices are taken from certain switched linear dynamical systems related to deadline misses.

Table: Examples from control systems with $d = 2$, ub: upper bound, mb: maximal size of SDP blocks, -: running time > 3600s, $*$: out of memory

m	n	lb	ub	Sparse ($d = 2$)		Dense ($d = 2$)		
				time (s)	ub	time (s)	ub	mb
2	6	0.9464	0.9782	0.42	0.9547	1.87	0.9539	21
3	8	0.7218	0.7467	0.60	0.7310	13.4	0.7305	36
4	10	0.7458	0.7738	0.75	0.7564	107	0.7554	55
5	12	0.8601	0.8937	1.08	0.8706	1157	0.8699	78
6	14	0.7875	0.8107	1.32	0.7958	-	-	-
7	16	1.1110	1.1531	1.81	1.1182	$*$	$*$	$*$
8	18	1.0487	1.0881	2.05	1.0569	$*$	$*$	$*$
9	20	0.7570	0.7808	2.52	0.7660	$*$	$*$	$*$
10	22	0.9911	1.0315	2.70	1.0002	$*$	$*$	$*$
11	24	0.7339	0.7530	3.67	0.7418	$*$	$*$	$*$

Main references

- Jie Wang, Haokun Li and Bican Xia, A New Sparse SOS Decomposition Algorithm Based on Term Sparsity, ISSAC'19, 2019.
- Jie Wang, Martina Maggio and Victor Magron, *SparseJSR: A Fast Algorithm to Compute Joint Spectral Radius via Sparse SOS Decompositions*, ACC'21, 2021.
- Jie Wang, Victor Magron and Jean B. Lasserre, *TSSOS: A Moment-SOS hierarchy that exploits term sparsity*, SIAM Optimization, 2021.
- Jie Wang, Victor Magron and Jean B. Lasserre, *Chordal-TSSOS: a moment-SOS hierarchy that exploits term sparsity with chordal extension*, SIAM Optimization, 2021.

- `SparseJSR`: https://github.com/wangjie212/SparseJSR

Thanks for your attention!