Switched Linear Systems and Infinite Products of Matrices


Pablo A. Parrilo

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology

Based on joint works with J. Altschuler (MIT),
A. Ahmadi (Princeton), R. Jungers (UCL), B. Legat (UCL)

SIAM AG 2019 - Bern


Motivation (from Optimization viewpoint)

Analysis of algorithms equivalent to analysis of dynamical systems:

E.g., for minimizing a quadratic function $f(x) = \frac{1}{2}x^T Ax$:

Iterative algorithm (e.g., gradient descent) $x_{k+1} = x_k - \gamma \nabla f(x_k)$	$\Longleftrightarrow$	Dynamical system (e.g., linear recursion) $x_{k+1} = (I - \gamma A)x_k$

Motivation (from Optimization viewpoint)

Analysis of algorithms equivalent to analysis of dynamical systems:

E.g., for minimizing a quadratic function $f(x) = \frac{1}{2}x^T Ax$:

Iterative algorithm (e.g., gradient descent) $x_{k+1} = x_k - \gamma \nabla f(x_k)$	$\Longleftrightarrow$	Dynamical system (e.g., linear recursion) $x_{k+1} = (I - \gamma A)x_k$

Q: What happens if we switch between algorithms?

Motivation (from Optimization viewpoint)

Analysis of algorithms equivalent to analysis of dynamical systems:

E.g., for minimizing a quadratic function $f(x) = \frac{1}{2}x^T Ax$:

Iterative algorithm (e.g., gradient descent) $x_{k+1} = x_k - \gamma \nabla f(x_k)$	$\Longleftrightarrow$	Dynamical system (e.g., linear recursion) $x_{k+1} = (I - \gamma A)x_k$

Q: What happens if we switch between algorithms?

For instance, one step of gradient, and one step of proximal mapping . . . ?

Motivation (from Optimization viewpoint)

Analysis of algorithms equivalent to analysis of dynamical systems:

E.g., for minimizing a quadratic function $f(x) = \frac{1}{2}x^T A x$:

Iterative algorithm (e.g., gradient descent) $x_{k+1} = x_k - \gamma \nabla f(x_k)$	$\Longleftrightarrow$	Dynamical system (e.g., linear recursion) $x_{k+1} = (I - \gamma A)x_k$

Q: What happens if we switch between algorithms?

For instance, one step of gradient, and one step of proximal mapping ...?

(or ADMM, or forward-backward, etc...)

Switching can be interesting

Example: Consider

$$A = \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix}$$

We have

$$\lim_{k\to\infty} \|A * A * A * \ldots\| = 0, \qquad \lim_{k\to\infty} \|B * B * B * \ldots\| = 0,$$

(in fact, A and B are nilpotent, so $A^2 = B^2 = 0$), but...

Switching can be interesting

Example: Consider

$$A = \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix}$$

We have

$$\lim_{k\to\infty} \|A * A * A * \ldots\| = 0, \qquad \lim_{k\to\infty} \|B * B * B * \ldots\| = 0,$$

(in fact, A and B are nilpotent, so $A^2 = B^2 = 0$), but...

$$\lim_{k\to\infty} \|A * B * A * B * \ldots\| = \infty \qquad (!)$$

Opportunity?

Perhaps we can also do this in reverse...?

Opportunity?

Perhaps we can also do this in reverse...?

Take Algorithm A (slow), and Algorithm B (bad).

Can we schedule them (e.g., alternate between them, or something else) to obtain a better/faster algorithm?

Opportunity?

Perhaps we can also do this in reverse...?

Take Algorithm A (slow), and Algorithm B (bad).

Can we schedule them (e.g., alternate between them, or something else) to obtain a better/faster algorithm?

Many, many possible variations.

Today, focus on analysis methods/tools, and a simple example.

Example

Consider a convex quadratic function $f(x) = \frac{1}{2}x^T Ax$, with $m \leqslant \lambda_i(A) \leqslant M$. For concreteness, we choose $m = 1$, $M = 5$.

Example

Consider a convex quadratic function $f(x) = \frac{1}{2}x^T Ax$, with $m \leqslant \lambda_i(A) \leqslant M$. For concreteness, we choose $m = 1$, $M = 5$.

Gradient method: $x_{k+1} = x_k - \alpha \nabla f(x_k) = (I - \alpha A)x_k$.
Optimal constant stepsize choice: $\alpha_\star = \frac{2}{m+M} = 1/3$, achieves rate $r_\star = \max_{\lambda \in [m,M]} |1 - \alpha_\star \lambda| = \frac{M-m}{M+m} = 2/3$.

Example

Consider a convex quadratic function $f(x) = \frac{1}{2}x^T Ax$, with $m \leqslant \lambda_i(A) \leqslant M$. For concreteness, we choose $m = 1$, $M = 5$.

Gradient method: $x_{k+1} = x_k - \alpha \nabla f(x_k) = (I - \alpha A)x_k$.
Optimal constant stepsize choice: $\alpha_\star = \frac{2}{m+M} = 1/3$, achieves rate $r_\star = \max_{\lambda \in [m,M]} |1 - \alpha_\star \lambda| = \frac{M-m}{M+m} = 2/3$.

Consider now two suboptimal methods, of stepsizes $\alpha_1 = 1/5$ and $\alpha_2 = 1/2$. The corresponding rates are $r_1 = 4/5$ and $r_2 = 3/2 > 1$ (divergent!).
Q: Can we schedule them to make them converge? Perhaps faster, even?

Example

Consider a convex quadratic function $f(x) = \frac{1}{2}x^T Ax$, with $m \leqslant \lambda_i(A) \leqslant M$. For concreteness, we choose $m = 1$, $M = 5$.

Gradient method: $x_{k+1} = x_k - \alpha \nabla f(x_k) = (I - \alpha A)x_k$.
Optimal constant stepsize choice: $\alpha_\star = \frac{2}{m+M} = 1/3$, achieves rate $r_\star = \max_{\lambda \in [m,M]} |1 - \alpha_\star \lambda| = \frac{M-m}{M+m} = 2/3$.

Consider now two suboptimal methods, of stepsizes $\alpha_1 = 1/5$ and $\alpha_2 = 1/2$. The corresponding rates are $r_1 = 4/5$ and $r_2 = 3/2 > 1$ (divergent!).
Q: Can we schedule them to make them converge? Perhaps faster, even?

Run them in "proportions" $(2/3, 1/3)$, e.g., $1 - 1 - 2 - 1 - 1 - 2 - \ldots$
The achieved rate is now $\frac{1}{\sqrt[3]{5}} \approx 0.5848$.
Better than both; actually outperforms the "optimal" constant stepsize!

Formal setting: switched linear systems

Given a finite set of matrices $\Sigma = \{A_1, \ldots, A_m\}$, consider the (switched) linear dynamical system:

$$x_{k+1} = A_{\sigma_k} x_k, \qquad \text{for } k = 0, 1, \ldots$$

where $\sigma_k \in \{1, \ldots, m\}$ and x_0 is a given initial state.

Formal setting: switched linear systems

Given a finite set of matrices $\Sigma = \{A_1, \ldots, A_m\}$, consider the (switched) linear dynamical system:

$$x_{k+1} = A_{\sigma_k} x_k, \qquad \text{for } k = 0, 1, \ldots$$

where $\sigma_k \in \{1, \ldots, m\}$ and x_0 is a given initial state.

Understanding this system is equivalent to analyzing the (left-)infinite matrix products

$$\cdots A_{\sigma_k} \cdots A_{\sigma_2} A_{\sigma_1}$$

How does one analyze/design this kind of things?

Different setups:

- **Deterministic** (worst-case) switching
  - Arbitrary switching, perhaps constrained (e.g., by an automaton)
  - Technical tool: Joint spectral radius
- **Random** switching
  - Probabilistic setup, could be i.i.d, or other process (e.g., Markov chain)
  - Technical tool: Lyapunov exponent

How does one analyze/design this kind of things?

Different setups:

- **Deterministic** (worst-case) switching
  - Arbitrary switching, perhaps constrained (e.g., by an automaton)
  - Technical tool: Joint spectral radius
- **Random** switching
  - Probabilistic setup, could be i.i.d, or other process (e.g., Markov chain)
  - Technical tool: Lyapunov exponent

Also, distinguish between

- **Analysis:** quantify convergence rate of given switching scheme σ
- **Synthesis:** design a switching scheme with good properties

Joint spectral radius

Given a set of matrices $\Sigma := \{A_1, \ldots, A_m\} \subset \mathbb{R}^{n \times n}$, what is the maximum "growth rate" that can be achieved by arbitrary switching?

$$\rho(\Sigma) := \limsup_{k \to \infty} \max_{\sigma \in \{1,\ldots,m\}^k} \|A_{\sigma_k} \cdots A_{\sigma_2} A_{\sigma_1}\|^{1/k}$$

Appeared in several different contexts: linear algebra (Rota-Strang 1960), wavelets (Daubechies-Lagarias 1992), switched linear systems, etc.

- If $m = 1$, "standard" spectral radius $\rho(A_1) = \max_i |\lambda_i|$.
- If $m \geqslant 2$, much more complicated...

Switching is hard...

Determining if $\rho(\Sigma) \leqslant 1$ is NP-hard (Tsitsiklis & Blondel 1997).

Switching is hard...

Determining if $\rho(\Sigma) \leqslant 1$ is NP-hard (Tsitsiklis & Blondel 1997).

$\rho(\Sigma)$ is not a semialgebraic function of the problem data.

Switching is hard...

Determining if $\rho(\Sigma) \leqslant 1$ is NP-hard (Tsitsiklis & Blondel 1997).

$\rho(\Sigma)$ is not a semialgebraic function of the problem data.

Determining if $\rho(\Sigma) \leqslant 1$ is undecidable (Blondel & Tsitsiklis 2000).

Switching is hard...

Determining if $\rho(\Sigma) \leqslant 1$ is NP-hard (Tsitsiklis & Blondel 1997).

$\rho(\Sigma)$ is not a semialgebraic function of the problem data.

Determining if $\rho(\Sigma) \leqslant 1$ is undecidable (Blondel & Tsitsiklis 2000).

Related work by Gurvits, Kozyakin, Barabanov, Wang-Lagarias "finiteness conjecture", Blondel-Nesterov, Theys, Jungers, etc.

Switching is hard...

Determining if $\rho(\Sigma) \leqslant 1$ is NP-hard (Tsitsiklis & Blondel 1997).

$\rho(\Sigma)$ is not a semialgebraic function of the problem data.

Determining if $\rho(\Sigma) \leqslant 1$ is undecidable (Blondel & Tsitsiklis 2000).

Related work by Gurvits, Kozyakin, Barabanov, Wang-Lagarias "finiteness conjecture", Blondel-Nesterov, Theys, Jungers, etc.

Still, how to compute/approximate $\rho(\Sigma)$?

What approximation guarantees can we have?

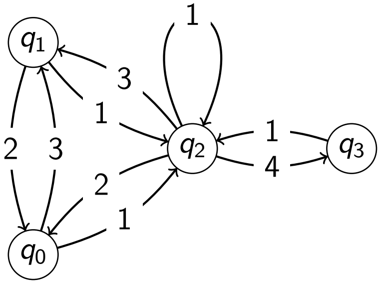
How to produce "bad" (high-growth) switching sequences?

Constrained switched systems

Switching constrained by a directed graph (automaton)

$$x_{k+1} = A_{\sigma_k} x_k$$

where $A_{\sigma_1}, A_{\sigma_2}, A_{\sigma_3}, A_{\sigma_4}, \ldots$ is a valid path.



Constrained Joint Spectral Radius:

$$\rho = \limsup_{k\to\infty} \max_{\sigma} \|A_{\sigma_k} \cdots A_{\sigma_1}\|^{1/k}.$$

where $\sigma_1, \ldots, \sigma_k$ is a path.

Bounding JSR using polynomials

Theorem

Let $p(x)$ be a strictly positive homogeneous polynomial of degree $2d$, that satisfies

$$p(A_i x) \leqslant \gamma^{2d} p(x), \qquad \forall x \in \mathbb{R}^n \quad i = 1, \dots, m$$

Then, $\rho(\Sigma) \leqslant \gamma$.

Bounding JSR using polynomials

Theorem

Let $p(x)$ be a strictly positive homogeneous polynomial of degree $2d$, that satisfies

$$p(A_i x) \leqslant \gamma^{2d} p(x), \qquad \forall x \in \mathbb{R}^n \quad i = 1, \dots, m$$

Then, $\rho(\Sigma) \leqslant \gamma$.

Proof: Since $p(x)$ is strictly positive, there exist $0 < \alpha \leqslant \beta$ such that

$$\alpha \|x\|^{2d} \leqslant p(x) \leqslant \beta \|x\|^{2d} \qquad \forall x \in \mathbb{R}^n.$$

and

$$\|A_{\sigma_k} \dots A_{\sigma_1}\| \leqslant \max_x \frac{\|A_{\sigma_k} \dots A_{\sigma_1} x\|}{\|x\|} \leqslant \left(\frac{\beta}{\alpha}\right)^{\frac{1}{2d}} \frac{p(A_{\sigma_k} \dots A_{\sigma_1} x)^{\frac{1}{2d}}}{p(x)^{\frac{1}{2d}}} \leqslant \left(\frac{\beta}{\alpha}\right)^{\frac{1}{2d}} \gamma^k.$$

The result follows by taking kth roots, and the limit $k \to \infty$.

Using sum of squares (SOS)

Want to find a strictly positive polynomial $p(x)$ that satisfies

$$p(x) \geqslant 0, \qquad \gamma^{2d} p(x) - p(A_i x) \geqslant 0, \quad i = 1, \ldots, m.$$

While this is not tractable, we can use instead

$$p(x) \text{ is SOS}, \qquad \gamma^{2d} p(x) - p(A_i x) \text{ is SOS}.$$

Then, $\rho(\Sigma) \leqslant \rho_{SOS} := \gamma$.

For fixed γ, this is an SOS program (convex optimization), can be solved using a semidefinite programming (SDP) solver. As degree $2d$ increases, better bounds.

Example

Based on (Ando and Shih 98). Consider the two matrices:

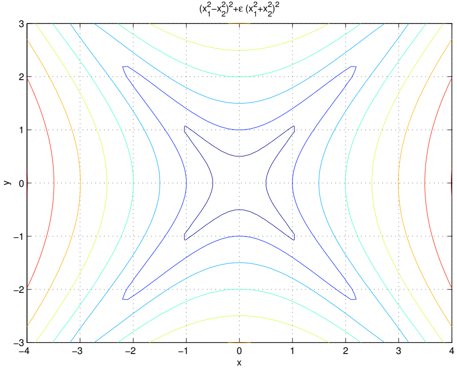
$$A_1 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix},$$

- The true spectral radius is $\rho(A_1, A_2) = 1$.
- A common quadratic Lyapunov function (i.e., $d = 2$), gives $\rho(A_1, A_2) \leqslant \sqrt{2}$.

Example (cont.)

A quartic SOS Lyapunov function is enough to prove an upper bound of $1+\epsilon$ for *every* $\epsilon > 0$, since

$$p(x) = (x_1^2 - x_2^2)^2 + \epsilon(x_1^2 + x_2^2)^2$$



satisfies

$$\begin{aligned}(1+\epsilon)p(x) - p(A_1x) &= (x_2^2 - x_1^2 + \epsilon(x_1^2 + x_2^2))^2\\(1+\epsilon)p(x) - p(A_2x) &= (x_1^2 - x_2^2 + \epsilon(x_1^2 + x_2^2))^2.\end{aligned}$$

SOS-based upper bound

Find γ, $p(x)$ such that

$$p(x) \text{ is SOS}, \qquad \gamma^{2d}\, p(x) - p(A_i x) \text{ is SOS}.$$

Let $\rho_{\mathrm{SOS},2d}$ be the smallest such γ.

What (if anything) can we say about its approximation properties?

SOS Guarantees

Theorem

The degree-2d SOS upper bound for $\rho(A_1, \ldots, A_m)$ *satisfies*

$$\eta^{-\frac{1}{2d}} \rho_{\mathrm{SOS},2d} \leqslant \rho \leqslant \rho_{\mathrm{SOS},2d},$$

where $\eta = \min\{m, \binom{n+d-1}{d}\}$.

- As degree $d \to \infty$, approximation ratio goes to 1 (!)
- For unbounded m, related to Barvinok/John's ellipsoid.
- For fixed m, proof based on Lyapunov iteration.

P.A. Parrilo and A. Jadbabaie, Approximation of the joint spectral radius using sum of squares, *Lin. Alg. Appl.*, 428(10), pp. 2385–2402, 2008.

SOS Guarantees

Theorem

The degree-2d SOS upper bound for $\rho(A_1, \ldots, A_m)$ satisfies

$$\eta^{-\frac{1}{2d}} \rho_{\mathrm{SOS},2d} \leqslant \rho \leqslant \rho_{\mathrm{SOS},2d},$$

where $\eta = \min\{m, \binom{n+d-1}{d}\}$*.*

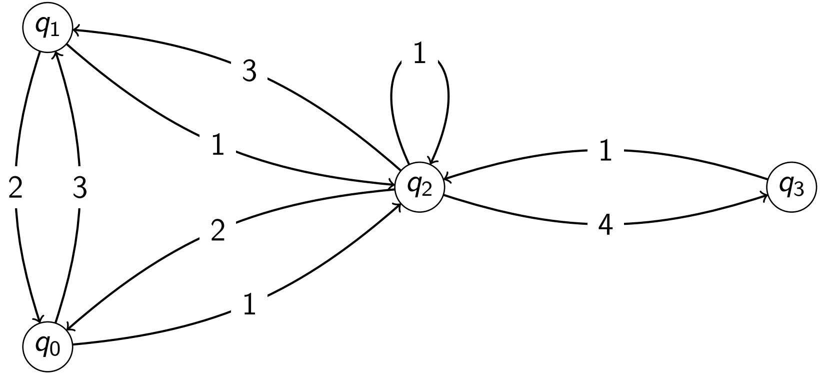
- As degree $d \to \infty$, approximation ratio goes to 1 (!)
- For unbounded m, related to Barvinok/John's ellipsoid.
- For fixed m, proof based on Lyapunov iteration.

But...

- How to produce "bad" (high-growth) switching sequences?

P.A. Parrilo and A. Jadbabaie, Approximation of the joint spectral radius using sum of squares, *Lin. Alg. Appl.*, 428(10), pp. 2385–2402, 2008.

Constrained switching and upper bounds for ρ



Use *local* polynomials: $u \to_\sigma v \qquad p_v(A_\sigma x) \leqslant \gamma^{2d} p_u(x).$

SOS Program

$$
\begin{aligned}
\text{minimize } & \gamma \\
p_v(x) \text{ is SOS} & \qquad \forall v \in V \\
\gamma^{2d} p_u(x) - p_v(A_\sigma x) \text{ is SOS} & \qquad \forall (u, v, \sigma) \in E.
\end{aligned}
$$

SOS Program

$$
\begin{aligned}
&\text{minimize } \gamma \\
&p_v(x) \text{ is SOS} && \forall v \in V \\
\gamma^{2d} p_u(x) - &p_v(A_\sigma x) \text{ is SOS} && \forall (u, v, \sigma) \in E.
\end{aligned}
$$

Q: What is the *dual*?

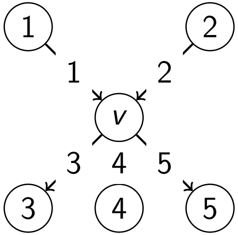
Dual: more in-flow than out-flow

Dual variables: pseudo-expectations $\widetilde{\mathbb{E}}[\cdot]$ for every *edge*, that satisfy

$$\sum_{(u,v,\sigma)\in E} \widetilde{\mathbb{E}}_{uv\sigma}[p(A_\sigma x)] \geqslant \gamma^{2d} \sum_{(v,w,\sigma)\in E} \widetilde{\mathbb{E}}_{vw\sigma}[p(x)], \qquad \forall v \in V, p(x) \text{ SOS},$$

Inequality between *incoming* and *outgoing* "probability flows":

$$\widetilde{\mathbb{E}}_{1v1}[p(A_1 x)] + \widetilde{\mathbb{E}}_{2v2}[p(A_2 x)] \geqslant \gamma^{2d}(\widetilde{\mathbb{E}}_{v33}[p(x)] + \widetilde{\mathbb{E}}_{v44}[p(x)] + \widetilde{\mathbb{E}}_{v55}[p(x)])$$



Building a high growth sequence

$$\sum_{(u,v,\sigma)\in E} \widetilde{\mathbb{E}}_{uv\sigma}[p(A_\sigma x)] \geqslant \gamma^{2d} \sum_{(v,w,\sigma)\in E} \widetilde{\mathbb{E}}_{vw\sigma}[p(x)], \qquad \forall v \in V, p(x) \text{ SOS},$$

Construct infinite sequence *backwards* using "best expectation," by lower bounding the maximum by the average.

Construction yields a guaranteed lower bound on JSR:

$$\frac{\gamma}{\left(d^{in}_{max}\right)^{\frac{1}{2d}}} \leqslant \lim_{k\to\infty} \|A_{\sigma_1} \cdots A_{\sigma_k}\|^{\frac{1}{k}}.$$

B. Legat, R. Jungers, P. Parrilo. "Generating unstable trajectories for switched systems via dual sum-of-squares techniques." HSCC2016, ACM, 2016.

The rank one case

Interesting special case: JSR of rank-one matrices ($A_i = u_i v_i^T$).

The rank one case

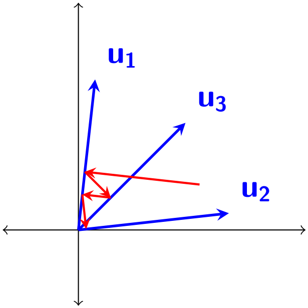
Interesting special case: JSR of rank-one matrices ($A_i = u_i v_i^T$).

Geometric picture: for simplicity, take A_i symmetric ($A_i = u_i u_i^T$).

Then the switched system

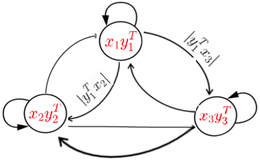
$$\begin{aligned} x_{k+1} &= A_{\sigma_k} x_k \\ &= u_{\sigma_k}(u_{\sigma_k}^T x_k) \end{aligned}$$

corresponds to projecting x_k onto lines u_{σ_k}.



JSR of rank one matrices as a graph problem

Given matrices $A_i = u_i v_i^T$ define a weighted directed graph $G = (E, V)$, where $E = [m]$ and $w_{ij} = \log |v_i^T u_j|$.



Optimal sequences can be obtained by solving the maximum cycle mean (MCM) problem for G, i.e., finding a directed simple cycle of largest average weight.

MCM can be efficiently solved using dynamic programming (Karp 1978).

Rank-one JSR is nice: efficiently solvable, optimal sequence always periodic (Ahmadi-P., also Gurvits-Samorodnitski).

A. Ahmadi, P. Parrilo. "Joint spectral radius of rank one matrices and the maximum cycle mean problem," *IEEE CDC*, 2012.

Random switching: Lyapunov exponent

Given a set of matrices $\Sigma := \{A_1, \ldots, A_m\}$ and a probability distribution $p \in \Delta_m$, what is the "growth rate" of random product of i.i.d. matrices?

$$R_p(\Sigma) := \lim_{k\to\infty} \|A_{\sigma_k} \cdots A_{\sigma_2} A_{\sigma_1}\|^{1/k}$$

- (Furstenberg-Kesten 1960) $R_p(\Sigma) = e^{\lambda_p(\Sigma)}$ a.s., where

$$\lambda_p(\Sigma) := \lim_{k\to\infty} \frac{1}{k} \mathbb{E}\left[\log \|A_{\sigma_k} \cdots A_{\sigma_2} A_{\sigma_1}\|\right]$$

- Characterization in terms of an invariant measure (hides complexity...)

Applications: ergodic theory, dynamical systems, fractals, stochastic linear systems, etc.

Random switching is hard...

Analysis: Given (Σ, p), **compute** convergence rate $R_p(\Sigma)$.

- Deciding stability (i.e. if $R_p(\Sigma) < 1$) is undecidable (Tsitsiklis & Blondel 1997).
- Still: how to compute/approximate? Special cases?
- Recently, nice convex bound (Sutter-Fawzi-Renner, `arXiv:1905.03270`)

Design: Given Σ, **optimize** convergence rate $\min_{p\in\Delta_m} R_p(\Sigma)$.

- Deciding stabilizability (i.e. if $\min_{p\in\Delta_m} R_p(\Sigma) < 1$) is NP-hard (Altschuler-P. 2019).
- Hard even for "simple" case of rank-one matrices, in contrast to analogous optimization for JSR!

Rank one case

Consider symmetric, rank-one matrices $\Sigma = \{A_i = u_i u_i^T\}_{i=1}^m$.

Analysis: Simple formula: $\lambda_p(\Sigma) = \sum_{ij=1}^m p_i p_j \log |u_i^T u_j|$.

- Ergodic formula: average time spent on edges of weighted graph
- Quadratic form on the simplex
- Computable, and in polynomial time.

Design: NP-hard to decide if $\min_{p\in\Delta_m} \lambda_p(\Sigma) < 0$. (i.e., if $R_p(\Sigma) < 1$).

- Reduction from Motzkin-Straus formulation of Independent Set, and use its hardness of approximation.

J. Altschuler, P. Parrilo. "Lyapunov Exponent of Rank One Matrices: Ergodic Formula and Inapproximability of the Optimal Distribution," arXiv:1905.07531.

Summary

- Switching is a powerful algorithmic tool, sometimes counterintuitive
- Key techniques: JSR and Lyapunov exponents
- SOS-based approximation for joint spectral radius
- Rank one case more tractable, but design problem still hard
- When is JSR/Lyapunov an algebraic function of data?

A. Ahmadi, P. Parrilo. "Joint spectral radius of rank one matrices and the maximum cycle mean problem," *IEEE CDC*, 2012.

A. Ahmadi, R. Jungers, P. Parrilo, M. Roozbehani. "Joint spectral radius and path-complete graph Lyapunov functions," *SIAM J. on Control and Optimization*, 52(1), 687–717, 2014.

J. Altschuler, P. Parrilo. "Lyapunov Exponent of Rank One Matrices: Ergodic Formula and Inapproximability of the Optimal Distribution," arXiv:1905.07531.

P.A. Parrilo and A. Jadbabaie, Approximation of the joint spectral radius using sum of squares, *Linear Algebra and its Applications*, 428(10), pp. 2385–2402, 2008.

B. Legat, R. Jungers, P. Parrilo. "Generating unstable trajectories for switched systems via dual sum-of-squares techniques." Proc. Hybrid Systems: Computation and Control (HSCC), ACM, 2016.