Devote this talk to

Scarlett's 2^{nd} Week Birthday

Sparse SOS Relaxation and Applications

Jiawang Nie

Institute for Mathematics and its Applications (IMA)

University of Minnesota

Joint work with Demmel and Sturmfels, UC Berkeley

BIRS, October 11, 2006

Outline of Talk

- Motivation

Outline of Talk

- Motivation
- Sparse SOS relaxations

Outline of Talk

- Motivation
- Sparse SOS relaxations
- Computational results and applications

Outline of Talk

- Motivation
- Sparse SOS relaxations
- Computational results and applications
- On the constrained optimization

Outline of Talk

- Motivation
- Sparse SOS relaxations
- Computational results and applications
- On the constrained optimization
- Conclusions

Motivation

- Let $f_1(x_{\Delta_1}), \cdots, f_m(x_{\Delta_m}) \in \mathbb{R}[x_1, \cdots, x_n]$

$$\Delta_1 \cup \cdots \cup \Delta_m = [n] := \{1, 2, \cdots, n\}$$

$$x_{\Delta_i} = \{x_k : k \in \Delta_i\}$$

- sparse polynomial optimization

$$\min_{x \in \mathbb{R}^n} \quad f(x) := \sum_{i=1}^{m} f_i(x_{\Delta_i})$$

- We want efficient SOS relaxations

SOS relaxation

- Dense SOS relaxation (Lasserre, Parrilo, ...)

$$f^*_{sos} := \max \quad \gamma \quad \text{s.t.} \quad f(x) - \gamma \ \text{ is SOS}$$

- Complexity bounds ($n = \#vars, \ \ 2d = \deg(f(x))$)
  - # variables $= \ \mathcal{O}(n^{2d})$
  - size of LMI $= \ \mathcal{O}(n^{d})$
  - comp. cost $= \ \mathcal{O}(n^{6d})$ (one IP iter.)

- Very expensive for medium or large size problems (e.g., $n = 30, 2d = 6$)

Outline of Talk

- Motivation
- Sparse SOS relaxations
- Computational results and applications
- On the constrained optimization
- Conclusions

Some Previous Work

- Finding the sparse support(Reznick, Parrilo, ...)

$$f(x) = \sum_i s_i^2(x) \Longrightarrow \quad \text{supp}(s_i(x)) \subset \ conv\{\frac{1}{2}\text{supp}(f(x))\}$$

Sometimes, this sparsity is too loose, e.g., $x_1^{2d} + \cdots + x_n^{2d}$

- Find the further sparse support (Muramatsu, Kojima, Kim, Waki, ...)
  - Use Cholesky factor of CSP matrix R
  - Can solve some large problems, but it might fail
  - Still expensive when R has dense Cholesky factor
- Convergent sparse SOS relaxations (Lasserre)
- Symmetry (Gaterman, Parrilo, Theobald, Scheiderer,...)

Sparse SOS relaxation

$$\min_{x \in \mathbb{R}^n} \quad f(x) := \sum_{i=1}^{m} f_i(x_{\Delta_i}) \qquad \|\Delta\| = \max_i |\Delta_i| \ll n$$

Sparse SOS relaxation

$$f_\Sigma^* := \max \quad \gamma \quad \textbf{s.t.} \quad f(x) - \gamma \in \sum_{i=1}^{m} s_i(x_{\Delta_i}) \in \sum \mathbb{R}_d[x_{\Delta_i}]^2.$$

The dual SDP is

$$\min_y \quad \sum_\alpha f_\alpha y_\alpha$$
$$s.t. \quad \mathcal{M}_d^{\Delta_i}(y) \succeq 0, \; i = 1, \cdots, m$$

Extracting Solutions

- Flat extension condition (Curto, Fialkow, Lasserre, Laurent, ...)
  rank$\mathcal{M}_d^{\Delta_i}(y) =$ rank$\mathcal{M}_{d-1}^{\Delta_i}(y)$
- For each Δ_i, find partial solution set $\mathcal{V}_i \subset \mathbb{R}^{\Delta_i}$, the projection $\mathbb{R}^n \to \mathbb{R}^{\Delta_i}$ of solutions
- If rank condition fails, apply random perturbation:

$$\min_{x \in \mathbb{R}^n} \quad f(x) := \sum_{i=1}^{m} f_i(x_{\Delta_i}) \quad + \quad \sum_{i=1}^{m} \xi_i x_i$$

where $\xi = (\xi_1, \cdots, \xi_n)$ is tiny and random

Complexity Comparison

- # variables $= \mathcal{O}(m\|\Delta\|^{2d})$
(for dense SOS $\mathcal{O}(n^{2d})$)
- size of LMI $= \mathcal{O}(\|\Delta\|^{d})$, m copies
(for dense SOS, one single $\mathcal{O}(n^{d})$)
- comp. cost $= \mathcal{O}(m^3\|\Delta\|^{6d})$ (one IP iter.)
(for dense SOS $\mathcal{O}(n^{6d})$)

Nonlinear least square problem

$$g_1(x_{\Delta_1}) = g_2(x_{\Delta_2}) = \cdots = g_m(x_{\Delta_m}) = 0$$

Set $f_i(x_{\Delta_i}) = g_i^2(x_{\Delta_i})$, and then solve

$$\min_{x \in \mathbb{R}^n} f(x) := \sum_{i=1}^{m} f_i(x_{\Delta_i})$$

- If the polynomial system is consistent,

$$f_{\Delta}^* = f_{sos}^* = f^* = 0$$

- This lower bound is not interesting
- Dual optimal solution y^* recovers the real zeros

One example

Consider the sparse polynomial system

$$2x_1^2 - 3x_1 + 2x_2 - 1 = 0$$
$$2x_i^2 + x_{i-1} - 3x_i + 2x_{i+1} - 1 = 0 \, (i = 2, \cdots, n-1)$$
$$2x_n^2 + x_{n-1} - 3x_n - 1 = 0.$$

Set it as an LS problem, and then solve by sparse SOS

$$\hat{x} = (1.8327, \, -0.1097, \, -0.5929, \cdots, -0.6658, \, -0.5960, \, -0.4164)$$
$$\tilde{x} = (-0.5708, \, -0.6819, \, -0.7025, \cdots, -0.6658, \, -0.5960, \, -0.4164).$$

A few seconds for $n = 100$

The sparse SOS might fail

Consider problem

$$\min_{x\in\mathbb{R}^3} \quad \underbrace{x_1^4 + (x_1x_2 - 1)^2}_{f_1(x_{\Delta_1})} + \underbrace{x_2^2x_3^2 + (x_3^2 - 1)^2}_{f_2(x_{\Delta_2})}$$

$$\Delta_1 = \{1,2\}, \qquad \Delta_2 = \{2,3\}$$

Numerical computation shows

$$f_\Delta^* \approx 5\cdot 10^{-5}$$

$$f_{sos}^* \approx 0.8499$$

$$f^* \approx 0.8650$$

$$f_\Delta^* < f_{sos}^* < f^*$$

Outline of Talk

- Motivation
- Sparse SOS relaxations
- Computational results and applications
- On the constrained optimization
- Conclusions

Randomly generated problems

Choose random $f(x) = \sum_i f_i(x_{\Delta_i})$ where

$$f_i(x_{\Delta_i}) = \mathbf{m}_d(x_{\Delta_i})^T \cdot A_i \cdot \mathbf{m}_d(x_{\Delta_i}) + b_i^T \mathbf{m}_{2d-1}(x_{\Delta_i}).$$

where Δ_i are chosen to be random subsets of $[n]$ such that $\|\Delta_i\| \leq \|\Delta\| \ll n$. Here A_i is

$$A_i = \sqrt{n} I_{N_i} + u_i u_i^T$$

This makes the global minimizers of $f(x)$ are contained in some compact set.

$$accuracy = \frac{|f(\hat{x}) - f_\Delta^*|}{\max\{1, |f(\hat{x})|\}}.$$

Computational Result (1)

	$\|\Delta\| = 3$				$\|\Delta\| = 4$			
	CPU seconds			accu	CPU seconds			accu
n	max	avr.	min	max	max	avr.	min	max
20	0.85	0.62	0.54	4.1e-9	1.46	1.15	0.91	2.4e-9
40	1.22	1.07	0.91	1.9e-9	2.86	2.49	2.25	2.9e-9
60	1.80	1.55	1.45	2.9e-9	4.43	4.17	3.91	3.1e-9
80	2.30	2.18	2.02	2.3e-9	6.26	5.94	5.24	3.7e-9
100	3.02	2.70	2.33	2.8e-9	7.85	7.41	7.01	5.0e-9

Table 1: Computational results for quartic polynomials with different sizes (deg = 4)

Computational Result (2)

	$\|\Delta\| = 3$				$\|\Delta\| = 4$			
	CPU seconds			accu	CPU seconds			accu
$2d$	max	avr	min	max	max	avr	min	max
4	1.01	0.87	0.77	2.5e-9	2.33	1.93	1.65	2.4e-9
6	3.22	2.96	2.67	1.8e-9	17.16	14.92	11.71	2.2e-9
8	13.07	11.44	10.13	1.7e-8	136.67	119.90	107.28	9.4e-8

Table 2: Computational results for polynomials of size $n = 30$ with different degrees

Solving Nonlinear Differential Equations

Consider the two-point boundary value problem (BVP)

$$F(t, x, x', x'') = 0, \;\; x(a) = \alpha, \; x(b) = \beta$$

where $F(t, x, x', x'') \in \mathbb{R}[t, x, x', x'']$. Use step size $h = \frac{b-a}{N+1}$ to discretize, then we get

$$F\left(t_k, x_k, \frac{x_{k+1} - x_{k-1}}{2h}, \frac{x_{k-1} - 2x_k + x_{k+1}}{h^2}\right) = 0, \; k = 1, \cdots, N$$

Solve this polynomial system by applying the sparse SOS relaxation method.

Boundary Value Problem (1)

$$x'' - 2x^3 = 0, \quad x(0) = \frac{1}{2},\ x(1) = \frac{1}{3}$$

N	eqn. error	$\|x_k - x(t_k)\|_\infty$	$\|x_k - x(t_k)\|_\infty / h^2$	time(sec)
20	5.2879e-07	1.5041e-05	6.6331e-003	1.18
30	2.6194e-07	1.9413e-05	1.8656e-002	2.09
40	3.0304e-07	4.3344e-05	7.2861e-002	3.99
50	6.5375e-07	1.5124e-04	3.9338e-001	6.82
60	1.5271e-06	4.8695e-04	1.8119e+00	7.77
70	1.2555e-06	5.2428e-04	2.6429e+00	9.16
80	9.7315e-07	6.1330e-04	4.0239e+00	9.78
90	2.7519e-06	1.9311e-03	1.5991e+01	10.81
100	1.8628e-06	8.1425e-04	8.3062e+00	8.79

Boundary Value Problem (2)

$$x'' + \frac{1}{2}(x+t)^3 = 0, \;\; x(0) = 0, \, x(1) = 0.$$

Use step size $h = \frac{1}{N+1}$ to discretize, then we get

$$2x_1 - x_2 + \frac{1}{2}h^2(x_1 + t_1)^3 = 0$$

$$2x_i - x_{i-1} - x_{i+1} + \frac{1}{2}h^2(x_i + t_i)^3 = 0, \, i = 2, \cdots, N-1$$

$$2x_N - x_{N-1} + \frac{1}{2}h^2(x_N + t_N)^3 = 0$$

For $N = 30$, we get the real solution in 2.5 CPU seconds

$$(-0.0159, -0.0312, -0.0459, -0.0600, -0.0735, -0.0864, -0.0985, -0.1099, -0.1205, -0.1302,$$
$$-0.1391, -0.1470, -0.1540, -0.1599, -0.1646, -0.1682, -0.1705, -0.1715, -0.1710, -0.1689,$$
$$-0.1651, -0.1596, -0.1521, -0.1425, -0.1307, -0.1164, -0.0995, -0.0796, -0.0567, -0.0302).$$

Sensor Network Localization

- Given Data:
  - anchors: $a_1, \cdots, a_m \in \mathbb{R}^d$
  - # sensors: n ($u_1, u_2, \cdots, u_n$ not given)
  - edge set: $\mathcal{A} = \{(i,j) : \ u_i \text{ is connected to } u_j\}$
    $\mathcal{B} = \{(i,k) : \ u_i \text{ is connected to } a_k\}$
  - distance: $d_{ij} \, ((i,j) \in \mathcal{A}), \ \ e_{ik} \, ((i,k) \in \mathcal{B})$
- Problem: Find sensors $u_1, u_2, \cdots, u_n \in \mathbb{R}^d$ such that

$$\|u_i - u_j\|_2 = d_{ij} \ \ \forall (i,j) \in \mathcal{A}$$

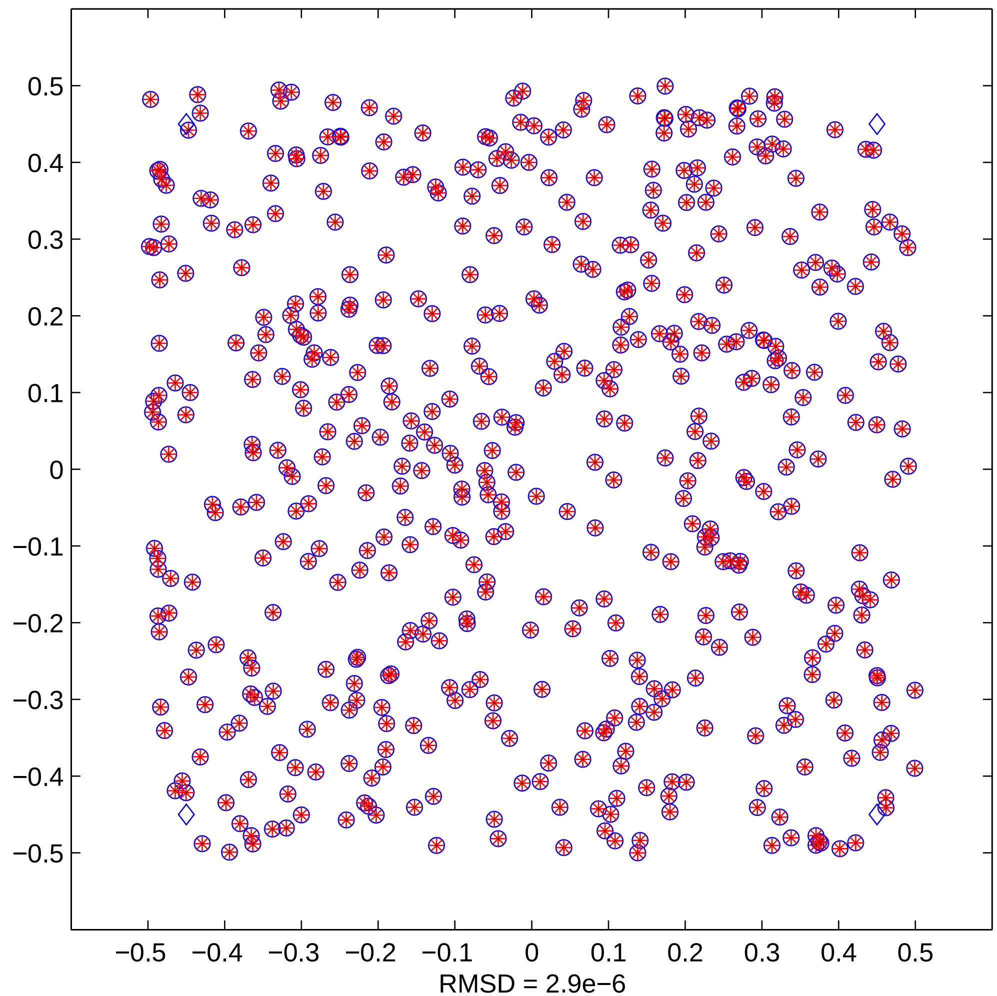
$$\|u_i - a_k\|_2 = e_{ik} \ \ \forall (i,k) \in \mathcal{B}$$

- NP-hard problem

Example

500 sensors, 4 anchors, range dist 0.3

This poly. system
has 1000 unknowns
and 11000 equations,
and was solved
in about 18 CPU mins



Perturbation Error Bound

$$f(X) = \sum_{(i,j)\in\mathcal{A}} \left(\|u_i - u_j\|_2^2 - d_{ij}^2\right)^2 + \sum_{(i,k)\in\mathcal{B}} \left(\|u_i - a_k\|_2^2 - e_{ik}^2\right)^2$$

Sparse SOS relax. returns true sensor location X^*

$$(d_{ij}, e_{ik}) \mapsto U^* = [u_1^*, \cdots, u_n^*]$$

$$U^* = \Psi((d_{ij}, e_{ik}))$$

Under some technical assump., it holds that

$$\underbrace{|\Psi((d_{ij}, e_{ik})) - \Psi((\hat{d}_{ij}, \hat{e}_{ik}))|}_{\text{sensor location error}} \le C \cdot \underbrace{\max\{|d_{ij} - \hat{d}_{ij}|, |e_{ik} - \hat{e}_{ik}|\}}_{\text{distance error}}$$

$$\varepsilon_U \le C \cdot \varepsilon$$

Outline of Talk

- Motivation
- Sparse SOS relaxations
- Computational results and applications
- On the constrained optimization
- Conclusions

Constrained Problem

$$f^* := \min_{x \in \mathbb{R}^n} \quad \sum_{i=1}^{m} f_i(x_{\Delta_i})$$

$$s.t. \quad g_{i,j}(x_{\Delta_i}) \geq 0,\ j \in \Omega_i,\ 1 \leq i \leq m$$

A nature sparse SOS relaxation is

$$f_N^* := \max \quad \gamma$$

$$s.t. \quad f(x) - \gamma - \sum_{i=1}^{m} \sum_{j \in \Omega_i} \sigma_{i,j}(x_{\Delta_i}) g_{i,j}(x_{\Delta_i}) \in \sum_i \sum \mathbb{R}_N[x_{\Delta_i}]^2$$

$$\deg(\sigma_{i,j} g_{i,j}) \leq 2N, \ \ \sigma_{i,j} \text{ is SOS},\ j \in \Omega_i,\ 1 \leq i \leq m.$$

Sparse Putinar's Theorem

Running Intersection Property (RIP):

$$\forall\, 1 \le i \le m,\ \exists\, k \le i-1 \ \text{ s.t. } \quad \Delta_i \cap (\cup_{j=1}^{i-1} \Delta_j) \subset \Delta_k.$$

Theorem (Lasserre): Under RIP and archimedean assumption, the lower bounds of sparse SOS relaxation converge to minimum

$$f_N^* \to f^*.$$

RIP is necessary

Running Intersection Property (RIP):

$$\forall\, 1 \le i \le m-1,\ \exists\, k \le i \ \text{ s.t. } \quad \Delta_{i+1} \cap \left(\cup_{j=1}^{i} \Delta_j\right) \subsetneq \Delta_k.$$

Without RIP, the convergence is not guaranteed, e.g.,

$$\min_{x \in \mathbb{R}^3} \quad \underbrace{\frac{1}{2}(x_1 + x_2)}_{f_1(x_{\Delta_1})} + \underbrace{\frac{1}{2}(x_1 + x_3)}_{f_2(x_{\Delta_2})} + \underbrace{\frac{1}{2}(x_2 + x_3)}_{f_3(x_{\Delta_2})}$$

$$s.t. \quad x_1^2 = x_1,\ x_2^2 = x_2,\ x_3^2 = x_3,$$

$$x_1 + x_2 \ge 1,\ x_1 + x_3 \ge 1, x_2 + x_3 \ge 1$$

$$\Delta_3 \cap \cup_{j=1}^{2} \Delta_j \not\subseteq \Delta_1, \quad \Delta_3 \cap \cup_{j=1}^{2} \Delta_j \not\subseteq \Delta_2.$$

Outline of Talk

- Motivation
- Sparse SOS relaxations
- Computational results and applications
- On the constrained optimization
- Conclusions

Conclusions

- Sparse SOS relax. is crucial for applications
- It might fail for particular problems
- It is important to write in “good” summation form

Conclusions

- Sparse SOS relax. is crucial for applications
- It might fail for particular problems
- It is important to write in “good” summation form
- Ongoing work (with Ranestad and Sturmfels)

  Algebraic degree of Semidefinite Programming

Conclusions

- Sparse SOS relax. is crucial for applications
- It might fail for particular problems
- It is important to write in “good” summation form
- Ongoing work (with Ranestad and Sturmfels)

  Algebraic degree of Semidefinite Programming

Thank you very much !

Devote this talk to

Scarlett's 2^{nd} Week Birthday