Sampling Algebraic Varieties for SOS Optimization

Diego Cifuentes

Laboratory for Information and Decision Systems
Electrical Engineering and Computer Science
Massachusetts Institute of Technology

Joint work with **Pablo A. Parrilo** (MIT)

arXiv:1511.06751

Coloquio Uniandes - 2017

Polynomial optimization on varieties

We consider a problem of the form

$$\begin{aligned} \min_{x} \quad & p(x) \\ \text{s.t.} \quad & x \in \mathcal{V} \end{aligned}$$

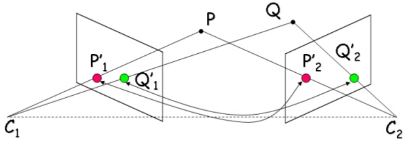
where $p \in \mathbb{R}[x]$ is a polynomial and $\mathcal{V} \subset \mathbb{R}^n$ is an *algebraic variety*.

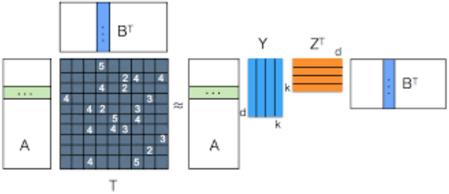
A variety is the zero set of some set of polynomial equations:

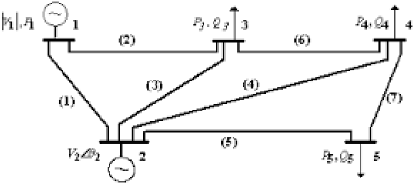
$$\mathcal{V} = \{x \in \mathbb{R}^n : h_j(x) = 0,\ 1 \leq j \leq m\}, \qquad h_j \in \mathbb{R}[x].$$

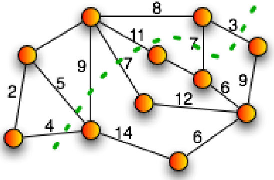
Examples: $SO(n)$, Grassmannian, Stiefel manifold, rank k tensors, $\{0,1\}^n$

Several applications: triangulation (vision), matrix completion, optimal power flow, low rank approximation, combinatorial optimization.









Sum-of-Squares (SOS)

For a polynomial $p \in \mathbb{R}[x]$ consider *deciding* nonnegavity

$$\text{is } p(x) \geq 0 \text{ for all } x \in \mathcal{V}?$$

This is computationally hard.

Tractable alternative: Convex relaxations based on *semidefinite programming* (SDP).

Sum-of-Squares (SOS)

For a polynomial $p \in \mathbb{R}[x]$ consider *deciding* nonnegavity

$$\text{is } p(x) \geq 0 \text{ for all } x \in \mathcal{V}?$$

This is computationally hard.

Tractable alternative: Convex relaxations based on *semidefinite programming* (SDP).

A *sufficient* condition is the existence of some $F \in \mathbb{R}[x]$ such that

$$p(z) = F(z) \text{ for all } z \in \mathcal{V} \qquad (\text{i.e., } p \equiv F \bmod I(\mathcal{V}))$$

$$F(x) \text{ is SOS} \qquad (\text{i.e., } F(x) = \sum_i f_i^2(x))$$

SOS($\mathcal{V}$) certificates

For a bound $d \in \mathbb{N}$, a *d-SOS($\mathcal{V}$) certificate* is an $F \in \mathbb{R}[x]$ s.t.

$$p(z) = F(z) \text{ for all } z \in \mathcal{V}, \quad F(x) \text{ is SOS}, \quad \deg(F) \leq d$$

SOS($\mathcal{V}$) certificates

For a bound $d \in \mathbb{N}$, a *d-SOS($\mathcal{V}$) certificate* is an $F \in \mathbb{R}[x]$ s.t.

$$p(z) = F(z) \text{ for all } z \in \mathcal{V}, \quad F(x) \text{ is SOS}, \quad \deg(F) \leq d$$

Computing SOS($\mathcal{V}$) certificates:

- Compute a Gröbner bases of $I(\mathcal{V})$.
- Find F using *semidefinite programming* (SDP) — polynomial time.
- In some cases we know a Gröbner basis (e.g., $\mathcal{V} = \{0, 1\}^n$).
- But it is typically very hard to find it.

SOS($\mathcal{V}$) certificates

For a bound $d \in \mathbb{N}$, a *d-SOS($\mathcal{V}$) certificate* is an $F \in \mathbb{R}[x]$ s.t.

$$p(z) = F(z) \text{ for all } z \in \mathcal{V}, \quad F(x) \text{ is SOS}, \quad \deg(F) \leq d$$

Equations SOS: relax the first condition to $p = F + \sum_j h_j g_j$, where $\mathcal{V} = \{x : h_j(x) = 0\}_j$. Although often used in practice,

- this approach is weaker than SOS($\mathcal{V}$).
- SDP is larger, e.g., PSD matrix size $\binom{n+d}{d} \gg \deg \mathcal{V} \binom{\dim \mathcal{V}+d}{d}$.

SOS($\mathcal{V}$) certificates

For a bound $d \in \mathbb{N}$, a *d-SOS($\mathcal{V}$) certificate* is an $F \in \mathbb{R}[x]$ s.t.

$$p(z) = F(z) \text{ for all } z \in \mathcal{V}, \quad F(x) \text{ is SOS}, \quad \deg(F) \leq d$$

This talk: a novel approach to compute SOS($\mathcal{V}$) certificates.

Sampling SOS

Def: A *sampling d-SOS precertificate* is a pair (F, Z) where $Z = \{z_1, \ldots, z_S\} \subset \mathcal{V}$ is a set of samples and $F \in \mathbb{R}[x]$ satisfies

$$p(z_s) = F(z_s) \quad \text{for } s = 1, \ldots, S, \quad F(x) \text{ is SOS}, \quad \deg(F) \leq d$$

It is a true certificate if furthermore $p(z) = F(z)$ for all $z \in \mathcal{V}$.

Sampling SOS

Def: A *sampling d-SOS precertificate* is a pair (F, Z) where $Z = \{z_1, \ldots, z_S\} \subset \mathcal{V}$ is a set of samples and $F \in \mathbb{R}[x]$ satisfies

$$p(z_s) = F(z_s) \quad \text{for } s = 1, \ldots, S, \quad F(x) \text{ is SOS}, \quad \deg(F) \leq d$$

It is a true certificate if furthermore $p(z) = F(z)$ for all $z \in \mathcal{V}$.

We can compute an SOS($\mathcal{V}$) certificate as follows:

1. Obtain generic (random) samples from the variety.
2. Given Z, compute a precertificate (F, Z) using an SDP.
3. Verify that (F, Z) is a certificate.

Sampling SOS (S-SOS)

Sampling SOS has the following features:

- The variety $\mathcal{V}$ is represented with a set of samples; no need to decide which equations $\{h_j\}$ to use. Also avoids multiplicities.

Sampling SOS (S-SOS)

Sampling SOS has the following features:

- The variety $\mathcal{V}$ is represented with a set of samples; no need to decide which equations $\{h_j\}$ to use. Also avoids multiplicities.
- The SDP is smaller, since it takes into account the structure of the coordinate ring $\mathbb{C}[\mathcal{V}]$.

Sampling SOS (S-SOS)

Sampling SOS has the following features:

- The variety $\mathcal{V}$ is represented with a set of samples; no need to decide which equations $\{h_j\}$ to use. Also avoids multiplicities.
- The SDP is smaller, since it takes into account the structure of the coordinate ring $\mathbb{C}[\mathcal{V}]$.
- Many interesting varieties are easy to sample ($SO(n)$, Grasmannians, rank k tensors, multiview variety), even if its defining equations (or Gröbner basis) are complicated.

Sampling SOS (S-SOS)

Sampling SOS has the following features:

- The variety $\mathcal{V}$ is represented with a set of samples; no need to decide which equations $\{h_j\}$ to use. Also avoids multiplicities.
- The SDP is smaller, since it takes into account the structure of the coordinate ring $\mathbb{C}[\mathcal{V}]$.
- Many interesting varieties are easy to sample ($SO(n)$, Grasmannians, rank k tensors, multiview variety), even if its defining equations (or Gröbner basis) are complicated.
- Integrates nicely with *Numerical Algebraic Geometry* (NAG). In particular, it can use straight-line-programs.

Computing SOS($\mathcal{V}$) certificates

1. Obtain generic (random) samples from the variety.
2. Given Z, compute a precertificate (F, Z) using an SDP.
3. Verify that (F, Z) is a certificate.

1. Generic samples of a variety

Many interesting varieties are easy to sample: $SO(n)$, Grassmannians, rank k tensors, multiview variety, secant varieties.

For instance, we can sample points in $SO(n)$ with the Cayley parametrization

$$A \mapsto (I - A)(I + A)^{-1}, \text{ for } A \text{ skew symmetric}$$

For an arbitrary $\mathcal{V}$, we can get generic samples using *Numerical Algebraic Geometry* (NAG). This offers several advantages over symbolic methods:

- naturally parellizable
- allow straight-line programs
- better numerical stability

1. Generic samples. How many?

Let d be a degree bound and $\mathcal{L}_d$ be the subspace of $\mathbb{C}[\mathcal{V}]$ up to degree d. We need $\dim(\mathcal{L}_d)/2$ samples (Hilbert series).

Thm: Let $\mathcal{V} = \mathcal{W} \cup \overline{W}$, with $\mathcal{W}$ irreducible. Let (F, Z) be an S-SOS pre-certificate with $\deg(F) \leq d$ and $|Z| \geq \dim(\mathcal{L}_d)/2$. If Z is generic, then (F, Z) is a certificate.

We can check if we have sufficient samples computing the rank of a matrix.

2. Sampling SDP

Given samples $Z = \{z_1, \ldots, z_S\} \subset \mathcal{V}$, compute pre-certificate (F, Z):

$$\begin{array}{ll} \text{find} & F \\ \text{s.t.} & F(z_s) = p(z_s) \text{ for } s = 1, \ldots, S, \qquad F(x) \text{ is SOS} \end{array}$$

2. Sampling SDP

Given samples $Z = \{z_1, \ldots, z_S\} \subset \mathcal{V}$, compute pre-certificate (F, Z):

$$\begin{array}{ll} \text{find} & F \\ \text{s.t.} & F(z_s) = p(z_s) \text{ for } s = 1, \ldots, S, \qquad F(x) \text{ is SOS} \end{array}$$

The first constraint is affine in F. The second, is a PSD constraint.

Proposition.

$F(x)$ is SOS iff it can be written as

$$F(x) = \boldsymbol{u}(x)^T Q \boldsymbol{u}(x), \quad Q \succeq 0$$

for some vector of monomials $\boldsymbol{u}(x) \in \mathbb{R}[x]^N$.

2. Sampling SDP

Given samples $Z = \{z_1, \ldots, z_S\} \subset \mathcal{V}$, compute pre-certificate (F, Z):

$$\begin{aligned} &\text{find} \quad F \\ &\text{s.t.} \quad F(z_s) = p(z_s) \text{ for } s = 1, \ldots, S, \qquad F(x) \text{ is SOS} \end{aligned}$$

The first constraint is affine in F. The second, is a PSD constraint.

Proposition.

$F(x)$ is SOS iff it can be written as

$$F(x) = \boldsymbol{u}(x)^T Q \boldsymbol{u}(x), \quad Q \succeq 0$$

for some vector of monomials $\boldsymbol{u}(x) \in \mathbb{R}[x]^N$.

Proof.

- If $Q \succeq 0$ then $Q =: V^T V$.

2. Sampling SDP

Given samples $Z = \{z_1, \dots, z_S\} \subset \mathcal{V}$, compute pre-certificate (F, Z):

$$\begin{aligned} \text{find} \quad & F \\ \text{s.t.} \quad & F(z_s) = p(z_s) \text{ for } s = 1, \dots, S, \qquad F(x) \text{ is SOS} \end{aligned}$$

The first constraint is affine in F. The second, is a PSD constraint.

Proposition.

$F(x)$ is SOS iff it can be written as

$$F(x) = \boldsymbol{u}(x)^T Q \boldsymbol{u}(x), \quad Q \succeq 0$$

for some vector of monomials $\boldsymbol{u}(x) \in \mathbb{R}[x]^N$.

Proof.

- If $Q \succeq 0$ then $Q =: V^T V$.
- Then $F(x) = \boldsymbol{f}(x)^T \boldsymbol{f}(x)$, where $\boldsymbol{f}(x) := V\boldsymbol{u}(x)$.

2. Sampling SDP

Given $Z \subset \mathcal{V}$ and a vector $\boldsymbol{u}(x) \in \mathbb{R}[x]^N$, the sampling SDP is

$$\boxed{\begin{array}{ll} \text{find} & Q \in \mathbb{R}^{N\times N}, \quad Q \succeq 0 \\ \text{s.t.} & Q \bullet \boldsymbol{u}(z_s)\boldsymbol{u}(z_s)^T = p(z_s), \ \text{for } s = 1, \ldots, S \end{array}}$$

Features:

- p can be a *straight-line* program.
- constraint matrices have *low rank*.
- we may *reduce complexity* by orthogonalizing $\boldsymbol{u}(x)$ w.r.t.

$$\langle f, g\rangle_Z := \textstyle\sum_{z_s\in Z}(f(z_s)g(\overline{z_s}) + f(\overline{z_s})g(z_s)).$$

Uses the structure of the coordinate ring $\mathbb{C}[\mathcal{V}]$.

2. Sampling SDP

Given $Z \subset \mathcal{V}$ and a vector $\boldsymbol{u}(x) \in \mathbb{R}[x]^N$, the sampling SDP is

$$\boxed{\begin{array}{ll} \text{find} & \quad Q \in \mathbb{R}^{N\times N}, \quad Q \succeq 0 \\ \text{s.t.} & Q \bullet \boldsymbol{u}(z_s)\boldsymbol{u}(z_s)^T = p(z_s), \;\; \text{for } s = 1, \ldots, S \end{array}}$$

Features:

- p can be a *straight-line* program.
- constraint matrices have *low rank*.
- we may *reduce complexity* by orthogonalizing $\boldsymbol{u}(x)$ w.r.t.

$$\langle f, g\rangle_Z := \textstyle\sum_{z_s\in Z}(f(z_s)g(\overline{z_s}) + f(\overline{z_s})g(z_s)).$$

Uses the structure of the coordinate ring $\mathbb{C}[\mathcal{V}]$.

Simple example: $SO(2)$

$p(X) = 4X_{21} - 2X_{11}X_{22} - 2X_{12}X_{21} + 3$ is nonnegative on $\mathcal{V} = SO(2)$.

Simple example: $SO(2)$

$p(X) = 4X_{21} - 2X_{11}X_{22} - 2X_{12}X_{21} + 3$ is nonnegative on $\mathcal{V} = SO(2)$.

Take 3 complex samples of $\mathcal{V}$

$$z_1 = \begin{bmatrix} -0.6+0.8i & 1.2+0.4i \\ -1.2-0.4i & -0.6+0.8i \end{bmatrix},\ z_2 = \begin{bmatrix} -1.2+0.4i & 0.6+0.8i \\ -0.6-0.8i & -1.2+0.4i \end{bmatrix},\ z_3 = \begin{bmatrix} -0.75+0.25i & 0.75+0.25i \\ -0.75-0.25i & -0.75+0.25i \end{bmatrix}.$$

Let $\boldsymbol{u}(x) = (1, X_{11}, X_{12}, X_{21}, X_{22})$ (5 terms). Orthogonalizing we get

$$\boldsymbol{u}^o(X) = (X_{21} + X_{22} - .8054,\ \ X_{21} - X_{22},\ \ X_{21} + X_{22} + 2.4831) \quad \text{(3 terms)}$$

Solving the SDP

$$\begin{aligned} \text{find} \qquad & Q \in \mathbb{R}^{3\times 3}, \quad Q \succeq 0 \\ \text{s.t.} \quad & p(z_s) = Q \bullet \boldsymbol{u}^o(z_s)\boldsymbol{u}^o(z_s)^T, \qquad \text{for } s = 1,2,3 \end{aligned}$$

we get $F(X) = (2X_{21} + 1)^2$.

3. Verifying S-SOS certificates

Verifying the validity of a pre-certificate (F, Z),
means testing if $g := p - F$ is identically zero on $\mathcal{V}$

3. Verifying S-SOS certificates

Verifying the validity of a pre-certificate (F, Z),
means testing if $g := p - F$ is identically zero on $\mathcal{V}$

This is the *polynomial identity testing* problem and there is a "probability-one" randomized algorithm:

consider a generic point z on each component of $\mathcal{V}$, and check if $g(z) = 0$.

Example: Nilpotent matrices

Let $\mathcal{V}$ be the variety of nilpotent matrices and $p(X) := \det(X + I)$.

Equations:

- (naive) The n^2 equations given by $X^n = 0$ have n^{n+1} terms!!!
- (smarter) We know a Gröbner basis ($\sim n!$ terms). Computing the normal form of $p(X)$ ($n!$ terms) is too hard!!!

Example: Nilpotent matrices

Let $\mathcal{V}$ be the variety of nilpotent matrices and $p(X) := \det(X + I)$.

Equations:

- (naive) The n^2 equations given by $X^n = 0$ have n^{n+1} terms!!!
- (smarter) We know a Gröbner basis ($\sim n!$ terms). Computing the normal form of $p(X)$ ($n!$ terms) is too hard!!!

Sampling:

- Easy to sample nilpotent matrices.
- For each sample X_s, we can evaluate $p(X_s)$ with Gaussian elimination.
- Since $p(X_s) = 1$ for all samples X_s, then $p(X) = (1)^2$ on the variety.

Example: Nilpotent matrices

Let $\mathcal{V}$ be the variety of nilpotent matrices and $p(X) := \det(X + I)$.

Equations:

- (naive) The n^2 equations given by $X^n = 0$ have n^{n+1} terms!!!
- (smarter) We know a Gröbner basis ($\sim n!$ terms). Computing the normal form of $p(X)$ ($n!$ terms) is too hard!!!

Sampling:

- Easy to sample nilpotent matrices.
- For each sample X_s, we can evaluate $p(X_s)$ with Gaussian elimination.
- Since $p(X_s) = 1$ for all samples X_s, then $p(X) = (1)^2$ on the variety.

Advantages:

- Avoid the problem of which equations to use (multiplicities).
- We can use straight-line programs (Gaussian elimination).
- Coordinate ring reduction.

Example: Orthogonal Procrustes

Weighted Orthog Procrustes

$$\begin{aligned} \min{}_X \quad & \|AXC - B\| \\ \text{s.t.} \quad & X^T X = I_k \\ & X \in \mathbb{R}^{n\times k} \end{aligned}$$

The sampling SDP is:

$$\begin{aligned} \max{}_{Q,\gamma} \quad & \gamma \\ \text{s.t.} \quad & \|AX_sC - B\|^2 - \gamma = Q \bullet \boldsymbol{u}(X_s)\boldsymbol{u}(X_s)^T \\ & Q \succeq 0 \end{aligned}$$

n	r	Equations SDP variables	Equations SDP constraints	Equations SDP time(s)	Gröbner basis (s)	Sampling SDP variables	Sampling SDP constraints	Sampling SDP time(s)
5	3	682	233	0.65	0.03	137	130	0.11
6	4	1970	576	1.18	9.94	326	315	0.14
7	5	4727	1207	3.56	-	667	651	0.24
8	6	9954	2255	13.88	-	1226	1204	0.45
9	7	19028	3873	42.14	-	2081	2052	1.10
10	8	33762	6238	124.43	-	3322	3285	2.48

Example: Cyclic 9-roots

Let $\mathcal{V} \subset \mathbb{C}^9$ be the positive dimensional part of the cyclic 9-roots problem

$$x_1 + x_2 + \cdots + x_8 + x_9 = 0$$
$$x_1x_2 + x_2x_3 + \cdots + x_8x_9 + x_9x_1 = 0$$
$$\vdots$$
$$x_1x_2x_3x_4x_5x_6x_7x_8 + \cdots + x_9x_1x_2x_3x_4x_5x_6x_7 = 0$$
$$x_1x_2x_3x_4x_5x_6x_7x_8x_9 = 1$$

Let's certify that $\mathcal{V} \cap \mathbb{R}^9 = \emptyset$ by showing that -1 is SOS on $\mathcal{V}$.

Example: Cyclic 9-roots

Let $\mathcal{V} \subset \mathbb{C}^9$ be the positive dimensional part of the cyclic 9-roots problem

$$
\begin{gathered}
x_1 + x_2 + \cdots + x_8 + x_9 = 0 \\
x_1x_2 + x_2x_3 + \cdots + x_8x_9 + x_9x_1 = 0 \\
\vdots \\
x_1x_2x_3x_4x_5x_6x_7x_8 + \cdots + x_9x_1x_2x_3x_4x_5x_6x_7 = 0 \\
x_1x_2x_3x_4x_5x_6x_7x_8x_9 = 1
\end{gathered}
$$

Let's certify that $\mathcal{V} \cap \mathbb{R}^9 = \emptyset$ by showing that -1 is SOS on $\mathcal{V}$.

Gröbner basis computation is complicated (M2 ran out of memory $\sim 5h$).

Sampling + NAG is simpler: Bertini gets generic samples in $2h\,45m$, and then we find an SOS($\mathcal{V}$) certificate in only $0.75s$.

Summary

- A new approach to SOS, that represents a variety with a generic set of samples (instead of some equations $h_j(x) = 0$)
- Takes advantage of coordinate ring reductions.
- Integrates SOS with NAG.

Summary

- A new approach to SOS, that represents a variety with a generic set of samples (instead of some equations $h_j(x) = 0$)
- Takes advantage of coordinate ring reductions.
- Integrates SOS with NAG.

If you want to know more:

- D. Cifuentes, P.A. Parrilo, Sampling algebraic varieties for sum of squares programs `arXiv:1511.06751`.

Gracias!