Dimension reduction for semidefinite programming

Pablo A. Parrilo

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

Joint work with **Frank Permenter** (MIT)
`arXiv:1608.02090`

CDC 2016 - Las Vegas

Semidefinite programs (SDPs)

$$\begin{array}{ll} \text{minimize} & \operatorname{Tr} CX \\ \text{subject to} & X \in \mathcal{A} \cap \mathbb{S}^n_+ \end{array}$$

Formulated over vector space $\mathbb{S}^n$ of $n \times n$ symmetric matrices.

- variable $X \in \mathbb{S}^n$
- $\mathcal{A} \subseteq \mathbb{S}^n$ an affine subspace, $C \in \mathbb{S}^n$ cost matrix
- $\mathbb{S}^n_+$ cone of psd matrices

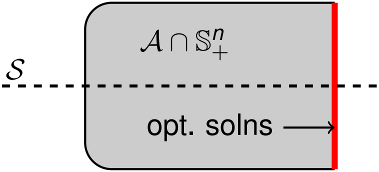
Efficiently solvable in theory; in practice, solving some instances impossible unless special structure is exploited.

Dimension reduction

Reformulate problem over subspace $\mathcal{S} \subseteq \mathbb{S}^n$ intersecting set of optimal solutions

$$\begin{array}{ll} \text{minimize} & \operatorname{Tr} CX \\ \text{subject to} & X \in \mathcal{A} \cap \mathbb{S}^n_+ \end{array} \qquad \begin{array}{ll} \text{minimize} & \operatorname{Tr} CX \\ \text{subject to} & X \in \mathcal{A} \cap \mathbb{S}^n_+ \cap \mathcal{S} \end{array}$$

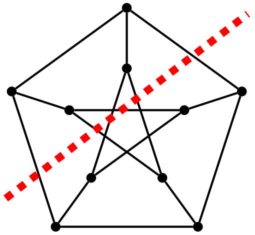
(Reformulation)



where $\mathbb{S}^n_+ \cap \mathcal{S}$ equals product $\mathcal{K}_i \times \cdots \times \mathcal{K}_m$ of 'simple' cones.

Reduction methods: *symmetry reduction* and *facial reduction*

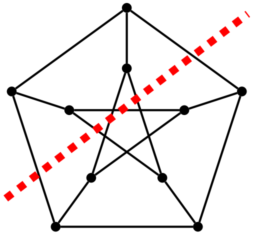
Symmetry reduction (MAXCUT relaxation example)



$$\begin{array}{ll} \text{minimize} & \operatorname{Tr} CX \\ \text{subject to} & X \in \mathcal{A} \cap \mathbb{S}^n_+ \end{array}$$

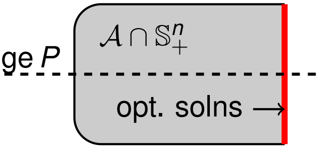
$\mathcal{A} := \{X \in \mathbb{S}^n : X_{ii} = 1\}$
$C :=$ adjacency matrix

Symmetry reduction (MAXCUT relaxation example)



$$\begin{array}{ll} \text{minimize} & \operatorname{Tr} CX \\ \text{subject to} & X \in \mathcal{A} \cap \mathbb{S}^n_+ \end{array}$$

$\mathcal{A} := \{X \in \mathbb{S}^n : X_{ii} = 1\}$
$C :=$ adjacency matrix

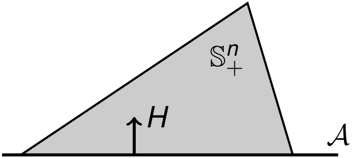


Idea: find special projection map P

- $P(X)$ optimal when X optimal.
- P explicitly constructed from automorphism group of graph.
- Range 'block-diagonal'—a direct-sum of matrix algebras.

(e.g., Schrijver '79; Gatermann-P. '03)

Facial reduction

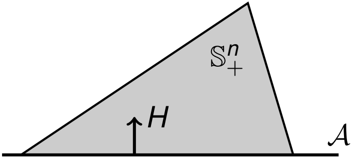


$$\begin{array}{ll} \text{minimize} & \operatorname{Tr} CX \\ \text{subject to} & X \in \mathcal{A} \cap \mathbb{S}^n_+ \end{array}$$

First, find *face* of $\mathbb{S}^n_+$ containing feasible set.

- There exists a hyperplane $H^\perp$ containing $\mathcal{A}$.
- $\mathbb{S}^n_+ \cap H^\perp$ a face—isomorphic to $\mathbb{S}^d_+$ for $d < n$.
- Face $\mathbb{S}^n_+ \cap H^\perp$ contains feasible set $\mathcal{A} \cap \mathbb{S}^n_+$.

Facial reduction



$$\begin{array}{ll} \text{minimize} & \operatorname{Tr} CX \\ \text{subject to} & X \in \mathcal{A} \cap \mathbb{S}^n_+ \end{array}$$

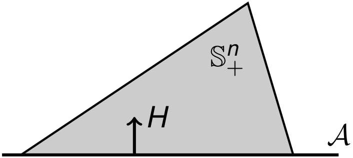
First, find *face* of $\mathbb{S}^n_+$ containing feasible set.

- There exists a hyperplane $H^\perp$ containing $\mathcal{A}$.
- $\mathbb{S}^n_+ \cap H^\perp$ a face—isomorphic to $\mathbb{S}^d_+$ for $d < n$.
- Face $\mathbb{S}^n_+ \cap H^\perp$ contains feasible set $\mathcal{A} \cap \mathbb{S}^n_+$.

Next, reformulate SDP over face:

$$\begin{array}{ll} \text{minimize} & \operatorname{Tr} CX \\ \text{subject to} & X \in \mathcal{A} \cap \mathbb{S}^n_+ \cap H^\perp \end{array}$$

Facial reduction



$$\begin{array}{ll} \text{minimize} & \operatorname{Tr} CX \\ \text{subject to} & X \in \mathcal{A} \cap \mathbb{S}^n_+ \end{array}$$

First, find *face* of $\mathbb{S}^n_+$ containing feasible set.

- There exists a hyperplane $H^\perp$ containing $\mathcal{A}$.
- $\mathbb{S}^n_+ \cap H^\perp$ a face—isomorphic to $\mathbb{S}^d_+$ for $d < n$.
- Face $\mathbb{S}^n_+ \cap H^\perp$ contains feasible set $\mathcal{A} \cap \mathbb{S}^n_+$.

Next, reformulate SDP over face:

$$\begin{array}{ll} \text{minimize} & \operatorname{Tr} CX \\ \text{subject to} & X \in \mathcal{A} \cap \mathbb{S}^n_+ \cap H^\perp \end{array}$$

Borwein-Wolkowicz '81; Pataki '00; Permenter-P. '14

Application specific approaches

Facial reduction:

- MAXCUT (Anjos, Wolkowicz)
- QAP (Zhao,Wolkowicz)
- Sums-of-squares optimization (Permenter-P., Waki-Muramatsu)
- Matrix completion (Krislock,Wolkowicz)
- ...

Symmetry reduction:

- MAXCUT (earlier example),
- QAP (de Klerk, Sotirov);
- Markov chains (Boyd *et al.*);
- codes (Schrijver; Laurent)
- ...

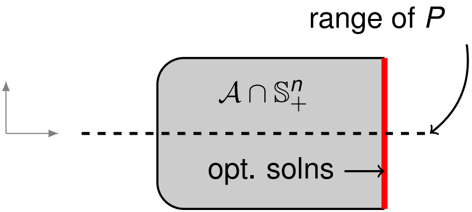
Our approach

This talk: new reduction method subsuming symmetry reduction

- Notion of ‘optimal’ reductions.
- A general purpose algorithm with optimality guarantees
- Jordan algebra interpretation; hence, easy extension to symmetric cone optimization (e.g., LP, SOCP).
- Combinatorial refinements for computational efficiency

How does symmetry reduction work?

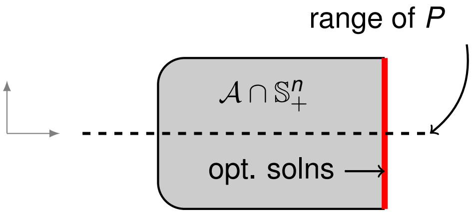
Given SDP $\min_{X \in \mathcal{A} \cap \mathbb{S}^n_+} \operatorname{Tr} CX$, method finds special orthogonal projection $P : \mathbb{S}^n \to \mathbb{S}^n$



If X feas./optimal, $P(X)$ feas./optimal.

How does symmetry reduction work?

Given SDP $\min_{X \in \mathcal{A} \cap \mathbb{S}^n_+} \operatorname{Tr} CX$, method finds special orthogonal projection $P : \mathbb{S}^n \to \mathbb{S}^n$



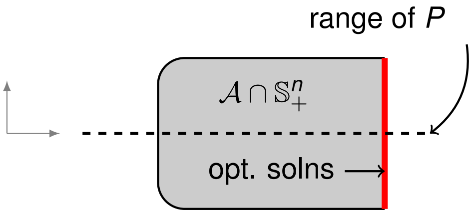
If X feas./optimal, $P(X)$ feas./optimal.

- P satisfies following conditions:

$$P(\mathcal{A}) \subseteq \mathcal{A}, \qquad P(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+, \qquad P(C) = C$$

How does symmetry reduction work?

Given SDP $\min_{X \in \mathcal{A} \cap \mathbb{S}^n_+} \operatorname{Tr} CX$, method finds special orthogonal projection $P : \mathbb{S}^n \to \mathbb{S}^n$



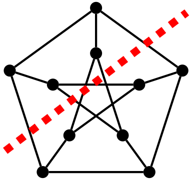
If X feas./optimal, $P(X)$ feas./optimal.

- P satisfies following conditions:

$$P(\mathcal{A}) \subseteq \mathcal{A}, \qquad P(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+, \qquad P(C) = C$$

- Hence, if X feasible then $P(X)$ feasible with equal cost:

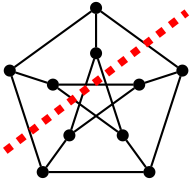
Example: a MAXCUT SDP relaxation



$$\begin{array}{ll} \text{minimize} & \operatorname{Tr} CX \\ \text{subject to} & X \in \mathcal{A} \cap \mathbb{S}^n_+ \end{array}$$

$\mathcal{A} := \{X \in \mathbb{S}^n : \ X_{ii} = 1\}$
$C :=$ adjacency matrix

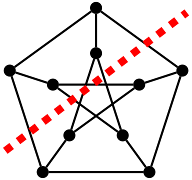
Example: a MAXCUT SDP relaxation



$$\begin{array}{ll} \text{minimize} & \operatorname{Tr} CX \\ \text{subject to} & X \in \mathcal{A} \cap \mathbb{S}^n_+ \end{array}$$

$\mathcal{A} := \{X \in \mathbb{S}^n : \ X_{ii} = 1\}$
$C :=$ adjacency matrix

Example: a MAXCUT SDP relaxation



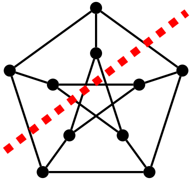
$$\begin{array}{ll} \text{minimize} & \operatorname{Tr} CX \\ \text{subject to} & X \in \mathcal{A} \cap \mathbb{S}^n_+ \end{array}$$

$\mathcal{A} := \{X \in \mathbb{S}^n : \ X_{ii} = 1\}$
$C :=$ adjacency matrix

Let $\mathcal{G}$ denote group of permutation matrices (automorphisms)

$$\mathcal{G} := \{U \text{ a permutation matrix } : U^T C U = C\}$$

Example: a MAXCUT SDP relaxation



$$\begin{array}{ll} \text{minimize} & \operatorname{Tr} CX \\ \text{subject to} & X \in \mathcal{A} \cap \mathbb{S}^n_+ \end{array}$$

$\mathcal{A} := \{X \in \mathbb{S}^n : \; X_{ii} = 1\}$
$C :=$ adjacency matrix

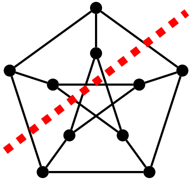
Let $\mathcal{G}$ denote group of permutation matrices (automorphisms)

$$\mathcal{G} := \{U \text{ a permutation matrix } : U^T C U = C\}$$

Taking $P(X) := \frac{1}{|\mathcal{G}|} \sum_{U \in \mathcal{G}} U^T X U$, desired conditions hold:

$$P(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+ \qquad P(\mathcal{A}) \subseteq \mathcal{A}, \qquad P(C) = C$$

Example: a MAXCUT SDP relaxation



$$\begin{array}{ll} \text{minimize} & \operatorname{Tr} CX \\ \text{subject to} & X \in \mathcal{A} \cap \mathbb{S}^n_+ \end{array}$$

$$\mathcal{A} := \{X \in \mathbb{S}^n : \; X_{ii} = 1\}$$
$$C := \text{adjacency matrix}$$

Let $\mathcal{G}$ denote group of permutation matrices (automorphisms)

$$\mathcal{G} := \{U \text{ a permutation matrix } : U^T C U = C\}$$

Taking $P(X) := \frac{1}{|\mathcal{G}|} \sum_{U \in \mathcal{G}} U^T X U$, desired conditions hold:

$$P(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+ \qquad P(\mathcal{A}) \subseteq \mathcal{A}, \qquad P(C) = C$$

Hence, range of P contains solutions: when X feasible, $P(X)$ feasible with equal cost.

Our approach: optimize over projections

Given SDP $\min_{X \in \mathcal{A} \cap \mathbb{S}^n_+} \langle C, X \rangle$, find map P that solves

$$
\begin{array}{ll}
\text{minimize} & \operatorname{rank} P \\
\text{subject to} & P(C) = C, P(I) = I \\
& P(\mathcal{A}) \subseteq \mathcal{A} \\
& P(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+ \\
& P : \mathbb{S}^n \to \mathbb{S}^n \text{ an orthogonal projection.}
\end{array}
$$

Our approach: optimize over projections

Given SDP $\min_{X \in \mathcal{A} \cap \mathbb{S}^n_+} \langle C, X \rangle$, find map P that solves

$$\begin{array}{ll} \text{minimize} & \operatorname{rank} P \\ \text{subject to} & P(C) = C, P(I) = I \\ & P(\mathcal{A}) \subseteq \mathcal{A} \\ & P(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+ \\ & P : \mathbb{S}^n \to \mathbb{S}^n \text{ an orthogonal projection.} \end{array}$$

Main properties:

- Can be solved in polynomial time (!)
- Range of P structured: a *Jordan subalgebra* of $\mathbb{S}^n$.
- $\mathbb{S}^n_+ \cap$ range P equals a product of symmetric cones.

Invariance characterization of optimal subspace

Theorem (Permenter-P.)

Orthogonal projection $P : \mathbb{S}^n \to \mathbb{S}^n$ solves

$$\begin{array}{ll} \textit{minimize} & \operatorname{rank} P \\ \textit{subject to} & P(C) = C, P(I) = I \\ & P(\mathcal{A}) \subseteq \mathcal{A} \\ & P(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+ \end{array}$$

iff the range of P solves

$$\begin{array}{ll} \textit{minimize} & \dim \mathcal{S} \\ \textit{subject to} & \mathcal{S} \ni \{I, X_{\mathcal{L}^\perp}, C\} \\ & \mathcal{S} \supseteq P_{\mathcal{L}}(\mathcal{S}) \\ & \mathcal{S} \supseteq \{X^2 : X \in \mathcal{S}\}, \end{array}$$

where $\mathcal{A} = X_{\mathcal{L}^\perp} + \mathcal{L}$, and $X_{\mathcal{L}^\perp}$ is the min-norm point of $\mathcal{A}$.

Subspace optimization and solution algorithm

$$\begin{array}{ll} \text{minimize} & \dim \mathcal{S} \\ \text{subject to} & \mathcal{S} \ni C, X_{\mathcal{L}^\perp}, I \\ & \mathcal{S} \supseteq P_{\mathcal{L}}(\mathcal{S}) \\ & \mathcal{S} \supseteq \{X^2 : X \in \mathcal{S}\} \end{array}$$

$\mathcal{S} \leftarrow \text{span}\{C, X_{\mathcal{L}^\perp}, I\}$
repeat
  $\mathcal{S} \leftarrow \mathcal{S} + P_{\mathcal{L}}(\mathcal{S})$
  $\mathcal{S} \leftarrow \mathcal{S} + \text{span}\{X^2 : X \in \mathcal{S}\}$
until converged.

Subspace optimization and solution algorithm

$$\begin{array}{ll} \text{minimize} & \dim \mathcal{S} \\ \text{subject to} & \mathcal{S} \ni C, X_{\mathcal{L}^\perp}, I \\ & \mathcal{S} \supseteq P_{\mathcal{L}}(\mathcal{S}) \\ & \mathcal{S} \supseteq \{X^2 : X \in \mathcal{S}\} \end{array}$$

$\mathcal{S} \leftarrow \text{span}\{C, X_{\mathcal{L}^\perp}, I\}$
repeat
 $\mathcal{S} \leftarrow \mathcal{S} + P_{\mathcal{L}}(\mathcal{S})$
 $\mathcal{S} \leftarrow \mathcal{S} + \text{span}\{X^2 : X \in \mathcal{S}\}$
until converged.

Properties of algorithm:

- Optimal subspace contains each iterate (induction)

Subspace optimization and solution algorithm

$$\begin{array}{ll} \text{minimize} & \dim \mathcal{S} \\ \text{subject to} & \mathcal{S} \ni C, X_{\mathcal{L}^\perp}, I \\ & \mathcal{S} \supseteq P_{\mathcal{L}}(\mathcal{S}) \\ & \mathcal{S} \supseteq \{X^2 : X \in \mathcal{S}\} \end{array}$$

$\mathcal{S} \leftarrow \text{span}\{C, X_{\mathcal{L}^\perp}, I\}$
repeat
 $\mathcal{S} \leftarrow \mathcal{S} + P_{\mathcal{L}}(\mathcal{S})$
 $\mathcal{S} \leftarrow \mathcal{S} + \text{span}\{X^2 : X \in \mathcal{S}\}$
until converged.

Properties of algorithm:

- Optimal subspace contains each iterate (induction)
- Computes ascending chain of subspaces—terminates.

Subspace optimization and solution algorithm

$$\begin{array}{ll} \text{minimize} & \dim \mathcal{S} \\ \text{subject to} & \mathcal{S} \ni C, X_{\mathcal{L}^\perp}, I \\ & \mathcal{S} \supseteq P_{\mathcal{L}}(\mathcal{S}) \\ & \mathcal{S} \supseteq \{X^2 : X \in \mathcal{S}\} \end{array}$$

$\mathcal{S} \leftarrow \text{span}\{C, X_{\mathcal{L}^\perp}, I\}$
repeat
  $\mathcal{S} \leftarrow \mathcal{S} + P_{\mathcal{L}}(\mathcal{S})$
  $\mathcal{S} \leftarrow \mathcal{S} + \text{span}\{X^2 : X \in \mathcal{S}\}$
until converged.

Properties of algorithm:

- Optimal subspace contains each iterate (induction)
- Computes ascending chain of subspaces—terminates.
- At termination, subspace feasible; hence, optimal.

Subspace optimization and solution algorithm

$$\begin{array}{ll} \text{minimize} & \dim \mathcal{S} \\ \text{subject to} & \mathcal{S} \ni C, X_{\mathcal{L}^\perp}, I \\ & \mathcal{S} \supseteq P_{\mathcal{L}}(\mathcal{S}) \\ & \mathcal{S} \supseteq \{X^2 : X \in \mathcal{S}\} \end{array}$$

$\mathcal{S} \leftarrow \mathrm{span}\{C, X_{\mathcal{L}^\perp}, I\}$
repeat
  $\mathcal{S} \leftarrow \mathcal{S} + P_{\mathcal{L}}(\mathcal{S})$
  $\mathcal{S} \leftarrow \mathcal{S} + \mathrm{span}\{X^2 : X \in \mathcal{S}\}$
until converged.

Properties of algorithm:

- Optimal subspace contains each iterate (induction)
- Computes ascending chain of subspaces—terminates.
- At termination, subspace feasible; hence, optimal.

Properties of minimization problem:

- Feasible set closed under intersection (lattice)

Subspace optimization and solution algorithm

$$\begin{array}{ll} \text{minimize} & \dim \mathcal{S} \\ \text{subject to} & \mathcal{S} \ni C, X_{\mathcal{L}^\perp}, I \\ & \mathcal{S} \supseteq P_{\mathcal{L}}(\mathcal{S}) \\ & \mathcal{S} \supseteq \{X^2 : X \in \mathcal{S}\} \end{array}$$

$\mathcal{S} \leftarrow \text{span}\{C, X_{\mathcal{L}^\perp}, I\}$
repeat
  $\mathcal{S} \leftarrow \mathcal{S} + P_{\mathcal{L}}(\mathcal{S})$
  $\mathcal{S} \leftarrow \mathcal{S} + \text{span}\{X^2 : X \in \mathcal{S}\}$
until converged.

Properties of algorithm:

- Optimal subspace contains each iterate (induction)
- Computes ascending chain of subspaces—terminates.
- At termination, subspace feasible; hence, optimal.

Properties of minimization problem:

- Feasible set closed under intersection (lattice)
- A unique solution.

Combinatorial descriptions

Great! Now, we can easily compute the optimal subspace $\mathcal{S}$.

Combinatorial descriptions

Great! Now, we can easily compute the optimal subspace $\mathcal{S}$.

But, often want/need additional properties (e.g., “dense” subspaces may not be very efficient).

Can tradeoff *dimension* with *sparsity of a basis*?

Combinatorial descriptions

Great! Now, we can easily compute the optimal subspace $\mathcal{S}$.

But, often want/need additional properties (e.g., “dense” subspaces may not be very efficient).

Can tradeoff *dimension* with *sparsity of a basis*?

Yes! Three kinds of sparse bases for $\mathcal{S}$:

- *Partition* subspaces: defined by a partition of $[n] \times [n]$.
- *Coordinate* subspaces: defined by a sparsity pattern
- *Combinatorial* subspaces: orthogonal basis of 0/1 matrices

E.g.,

$$\begin{bmatrix} a & a & b \\ a & a & b \\ b & b & c \end{bmatrix} \quad \text{vs.} \quad \begin{bmatrix} a & b & 0 \\ b & c & 0 \\ 0 & 0 & d \end{bmatrix} \quad \text{vs.} \quad \begin{bmatrix} a & 0 & b \\ 0 & a & c \\ b & c & b \end{bmatrix}$$

Finding optimal structured subspaces

The main algorithm can be adapted to compute the optimal subspace for each of these three cases.

Finding optimal structured subspaces

The main algorithm can be adapted to compute the optimal subspace for each of these three cases.

Key property (again): lattice structure (closedness under intersection)

Finding optimal structured subspaces

The main algorithm can be adapted to compute the optimal subspace for each of these three cases.

Key property (again): lattice structure (closedness under intersection)

E.g., for partition subspaces, instead of optimizing over lattice of subspaces, use the lattice of partitions:

$$
\begin{array}{ll}
\text{minimize} & \dim \mathcal{S} \\
\text{subject to} & \mathcal{S} \ni C, X_{\mathcal{L}^\perp}, I \\
& \mathcal{S} \supseteq P_{\mathcal{L}}(\mathcal{S}) \\
& \mathcal{S} \supseteq \{X^2 : X \in \mathcal{S}\} \\
& \mathcal{S} \text{ is a partition subspace}
\end{array}
$$

$\mathcal{P} \leftarrow \text{Part}\{C, X_{\mathcal{L}^\perp}, I\}$

repeat

  $\mathcal{P} \leftarrow \text{refine}(\mathcal{P}, P_{\mathcal{L}})$

  $\mathcal{P} \leftarrow \text{refine}(\mathcal{P}, X \mapsto X^2)$

until converged.

Finding optimal structured subspaces

The main algorithm can be adapted to compute the optimal subspace for each of these three cases.

Key property (again): lattice structure (closedness under intersection)

E.g., for partition subspaces, instead of optimizing over lattice of subspaces, use the lattice of partitions:

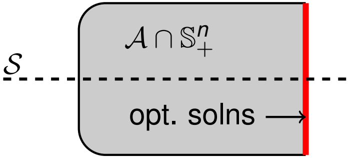
$$
\begin{array}{ll}
\text{minimize} & \dim \mathcal{S} \\
\text{subject to} & \mathcal{S} \ni C, X_{\mathcal{L}^\perp}, I \\
& \mathcal{S} \supseteq P_{\mathcal{L}}(\mathcal{S}) \\
& \mathcal{S} \supseteq \{X^2 : X \in \mathcal{S}\} \\
& \mathcal{S} \text{ is a partition subspace}
\end{array}
$$

$\mathcal{P} \leftarrow \text{Part}\{C, X_{\mathcal{L}^\perp}, I\}$
repeat
  $\mathcal{P} \leftarrow \text{refine}(\mathcal{P}, P_{\mathcal{L}})$
  $\mathcal{P} \leftarrow \text{refine}(\mathcal{P}, X \mapsto X^2)$
until converged.

Great! But there's more...

Decomposition via Jordan algebras

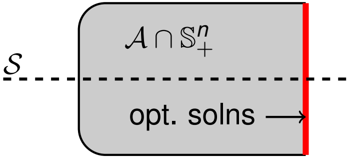
Given SDP $\min_{X \in \mathcal{A} \cap \mathbb{S}^n_+} \langle C, X \rangle$, we've found a subspace invariant under $X \mapsto X^2$ containing optimal solutions:



$$\mathcal{S} \supseteq \{X^2 : X \in \mathcal{S}\}$$

Decomposition via Jordan algebras

Given SDP $\min_{X \in \mathcal{A} \cap \mathbb{S}^n_+} \langle C, X \rangle$, we've found a subspace invariant under $X \mapsto X^2$ containing optimal solutions:



$$\mathcal{S} \supseteq \{X^2 : X \in \mathcal{S}\}$$

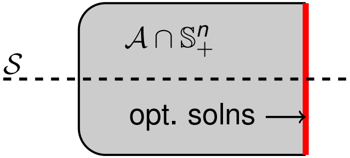
- Subspaces invariant under $X \mapsto X^2$ have decomposition

$$\mathcal{S} = Q \begin{pmatrix} \mathcal{S}_1 & 0 & \dots & 0 \\ 0 & \mathcal{S}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \mathcal{S}_m \end{pmatrix} Q^T,$$

$\mathcal{S}_i$ are *simple Jordan algebras*

Decomposition via Jordan algebras

Given SDP $\min_{X \in \mathcal{A} \cap \mathbb{S}^n_+} \langle C, X \rangle$, we've found a subspace invariant under $X \mapsto X^2$ containing optimal solutions:



$$\mathcal{S} \supseteq \{X^2 : X \in \mathcal{S}\}$$

- Subspaces invariant under $X \mapsto X^2$ have decomposition

$$\mathcal{S} = Q \begin{pmatrix} \mathcal{S}_1 & 0 & \dots & 0 \\ 0 & \mathcal{S}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \mathcal{S}_m \end{pmatrix} Q^T,$$

$\mathcal{S}_i$ are *simple Jordan algebras*

- Number of distinct eigenvalues of generic element equals *rank* of $\mathcal{S}_i$—a complexity measure.

Minimizing dimension optimizes decomposition

$$\begin{array}{ll} \text{minimize} & \dim \mathcal{S} \\ \text{subject to} & \mathcal{S} \ni X_{\mathcal{L}^\perp}, C, I \\ & \mathcal{S} \supseteq P_{\mathcal{L}}(\mathcal{S}) \\ & \mathcal{S} \supseteq \{X^2 : X \in \mathcal{S}\}, \end{array}$$

All feasible subspaces have decomp. $\mathcal{S} = \oplus_{i=1}^{d_\mathcal{S}} \mathcal{S}_i$. In what sense does solution $\mathcal{S}^*$ optimize the ranks of each $\mathcal{S}_i$?

Minimizing dimension optimizes decomposition

$$\begin{array}{ll} \text{minimize} & \dim \mathcal{S} \\ \text{subject to} & \mathcal{S} \ni X_{\mathcal{L}^\perp}, C, I \\ & \mathcal{S} \supseteq P_{\mathcal{L}}(\mathcal{S}) \\ & \mathcal{S} \supseteq \{X^2 : X \in \mathcal{S}\}, \end{array}$$

All feasible subspaces have decomp. $\mathcal{S} = \oplus_{i=1}^{d_{\mathcal{S}}} \mathcal{S}_i$. In what sense does solution $\mathcal{S}^*$ optimize the ranks of each $\mathcal{S}_i$?

Thm. (Permenter-P.):

- $\mathcal{S}^*$ minimizes $\sum_i \operatorname{rank} \mathcal{S}_i$ and $\max_i \operatorname{rank} \mathcal{S}_i$

Minimizing dimension optimizes decomposition

$$\begin{array}{ll} \text{minimize} & \dim \mathcal{S} \\ \text{subject to} & \mathcal{S} \ni X_{\mathcal{L}^\perp}, C, I \\ & \mathcal{S} \supseteq P_{\mathcal{L}}(\mathcal{S}) \\ & \mathcal{S} \supseteq \{X^2 : X \in \mathcal{S}\}, \end{array}$$

All feasible subspaces have decomp. $\mathcal{S} = \oplus_{i=1}^{d_\mathcal{S}} \mathcal{S}_i$. In what sense does solution $\mathcal{S}^*$ optimize the ranks of each $\mathcal{S}_i$?

Thm. (Permenter-P.):

- $\mathcal{S}^*$ minimizes $\sum_i \operatorname{rank} \mathcal{S}_i$ and $\max_i \operatorname{rank} \mathcal{S}_i$
- *Majorization* inequalities hold, i.e., for each $m \geq 1$

$$\sum_{i=1}^{m} \operatorname{rank} \mathcal{S}_i^* \leq \sum_{i=1}^{m} \operatorname{rank} \mathcal{S}_i$$

(ranks sorted in decreasing order)

Majorization example

Subspaces (parametrized by u_i and v_i) and their rank vectors

$$\begin{pmatrix} u_1 & u_2 & 0 & 0 & 0 \\ u_2 & u_3 & 0 & 0 & 0 \\ 0 & 0 & u_4 & 0 & 0 \\ 0 & 0 & 0 & u_5 & u_6 \\ 0 & 0 & 0 & u_6 & u_7 \end{pmatrix} \qquad \begin{pmatrix} v_1 & v_2 & 0 & 0 & 0 \\ v_2 & v_3 & 0 & 0 & 0 \\ 0 & 0 & v_4 & v_5 & v_6 \\ 0 & 0 & v_5 & v_7 & v_8 \\ 0 & 0 & v_6 & v_8 & v_9 \end{pmatrix}$$

$$r_u = (2, 1, 2) \qquad r_v = (2, 3)$$

Majorization example

Subspaces (parametrized by u_i and v_i) and their rank vectors

$$\begin{pmatrix} u_1 & u_2 & 0 & 0 & 0 \\ u_2 & u_3 & 0 & 0 & 0 \\ 0 & 0 & u_4 & 0 & 0 \\ 0 & 0 & 0 & u_5 & u_6 \\ 0 & 0 & 0 & u_6 & u_7 \end{pmatrix} \qquad \begin{pmatrix} v_1 & v_2 & 0 & 0 & 0 \\ v_2 & v_3 & 0 & 0 & 0 \\ 0 & 0 & v_4 & v_5 & v_6 \\ 0 & 0 & v_5 & v_7 & v_8 \\ 0 & 0 & v_6 & v_8 & v_9 \end{pmatrix}$$

$$r_u = (2, 1, 2) \qquad r_v = (2, 3)$$

Vector $r'_u = (2, 2, 1)$ majorized by $r'_v = (3, 2, 0)$:

$$2 \leq 3, \qquad 2 + 2 \leq 3 + 2, \qquad 2 + 2 + 1 \leq 3 + 2 + 0$$

Jordan algebras

- Jordan algebras are commutative algebras satisfying *Jordan identity*
$$(X \circ Y) \circ X^2 = X \circ (Y \circ X^2)$$
- The vector space $\mathbb{S}^n$ a Jordan algebra if equipped with product
$$X \circ Y := \frac{1}{2}(XY + YX)$$
- The *subalgebras* of $\mathbb{S}^n$ precisely the sets closed under squaring map $X \mapsto X^2$ since
$$XY + YX = (X + Y)^2 - X^2 - Y^2.$$
- Structure theorem of Jordan-von Neumann-Wigner describes subalgebras of $\mathbb{S}^n$....

Decomposition of $\mathcal{S} \cap \mathbb{S}^n_+$

If $\mathcal{S} \subset \mathbb{S}^n$ a Jordan subalgebra, it equals direct-sum $\oplus_{i=1}^m \mathcal{S}_i$, where each $\mathcal{S}_i$ is isomorphic to one of the following:

- Algebra of Hermitian matrices with real, complex or quaternion entries
- A spin-factor algebra

Implies *cone-of-squares* $\mathcal{S} \cap \mathbb{S}^n_+$ isomorphic to product of

- PSD cones with real/complex/quaternion entries
- Lorentz cones

Yields reformulation of original SDP over this product

$$\begin{array}{ll} \text{minimize} & \operatorname{Tr} CX \\ \text{subject to} & X \in \mathcal{A} \cap \mathbb{S}^n_+ \end{array} \qquad \begin{array}{ll} \text{minimize} & \operatorname{Tr} CX \\ \text{subject to} & X \in \mathcal{A} \cap \underbrace{T(\mathcal{K}_1 \times \cdots \times \mathcal{K}_m)}_{\mathcal{S} \cap \mathbb{S}^n_+} \end{array}$$

Computational results

Comparison with reduction method of de Klerk '10 survey (generating *-algebras from data):

instance	$\mathcal{S}^*$	$\mathcal{S}_{data}$
hamming_7_5_6	5	8256
hamming_8_3_4	5	32896
hamming_9_5_6	6	131328
hamming_9_8	6	131328
hamming_10_2	7	524800

- Table list dimension of our subspace $\mathcal{S}^* \subseteq \mathbb{S}^n$ and subspace $\mathcal{S}_{data} \subseteq \mathbb{S}^n$ found by generating *-algebra.
- Decomposing $\mathcal{S}^*$ yields a linear program.

Results: SOSOPT (Seiler '13) Demo scripts

Script Name	n (before)	n (after)
sosoptdemo2	13, 3	$3, 2 \times 3, 1 \times 7$
sosoptdemo4	35	$5 \times 5, 1 \times 10$
gsosoptdemo1	9, 5	$6, 3 \times 2, 2$
IOGainDemo_3	15, 8	$10, 5 \times 2, 3$
Chesi(1|2)_IterationWithVlin	9, 5	$6, 3 \times 2, 2$
Chesi3_GlobalStability	14, 5	8, 6, 3, 2
Chesi(3|4)_IterationWithVlin	9, 5	$6, 3 \times 2, 2$
Chesi(5|6)_Bootstrap	19, 9	$13, 6 \times 2, 3$
Chesi(5|6)_IterationWithVlin	19, 9	$13, 6 \times 2, 3$
Coutinho3_IterationWithVlin	9, 5	$6, 3 \times 2, 2$
HachichoTibken_Bootstrap	19, 9	12, 7, 6, 3
HachichoTibken_IterationWithVlin	19, 9	12, 7, 6, 3
Hahn_IterationWithVlin	9, 5	6, 3, 3, 2
KuChen_IterationWithVlin	19, 9	$13, 6 \times 2, 3$
Parrilo1_GlobalStabilityWithVec	3, 2	$2, 1 \times 3$
Parrilo2_GlobalStabilityWithMat	3, 2	$2, 1 \times 3$
VDP_IterationWithVball	5, 4	$3 \times 2, 2, 1$
VDP_IterationWithVlin	9, 5	$6, 3 \times 2, 2$
VDP_LinearizedLyap	9, 5	$6, 3 \times 2, 2$
VannelliVidyasagar2_Bootstrap	19, 9	$13, 6 \times 2, 3$
VannelliVidyasagar2_IterationWithVlin	19, 9	$13, 6 \times 2, 3$
VincentGrantham_IterationWithVlin	9, 5	$6, 3 \times 2, 2$
WTBenchmark_IterationWithVlin	19, 9	$13, 6 \times 2, 3$

Conclusions

New reduction method for SDP.

- Generalizes symmetry reduction and *-algebra-methods
- Fully algorithmic, don't need to compute automorphisms!
- Yields optimal 'block-diagonalization' (majorization)
- Can exploit combinatorial description of subspace
- Through Jordan algebra theory, extends to LP/SOCP/...

Conclusions

New reduction method for SDP.

- Generalizes symmetry reduction and *-algebra-methods
- Fully algorithmic, don't need to compute automorphisms!
- Yields optimal 'block-diagonalization' (majorization)
- Can exploit combinatorial description of subspace
- Through Jordan algebra theory, extends to LP/SOCP/...

Preprint at `arXiv:1608.02090`.

Conclusions

New reduction method for SDP.

- Generalizes symmetry reduction and *-algebra-methods
- Fully algorithmic, don't need to compute automorphisms!
- Yields optimal 'block-diagonalization' (majorization)
- Can exploit combinatorial description of subspace
- Through Jordan algebra theory, extends to LP/SOCP/...

Preprint at `arXiv:1608.02090`.

Thanks for your attention!