Exploiting Structured Sparsity in Large Scale Semidefinite Programming Problems

M. Kojima

Tokyo Institute of Technology

International Congress of Mathematical Software 2010

Kobe University, Kobe, Japan

September 13 - 17, 2010

- Kim, Kojima, Mevissen and Yamashita, “Exploiting sparsity in linear and nonlinear inequalities via positive semidefinite matrix completion”, *Mathematical Programming* to appear.

Outline

Outline

0 Semidefinite Programming (SDP)

1 A simple example for 2 types of sparsities

2 Chordal graph

3 Domain-space sparsity

4 Range-space sparsity

5 Numerical results

6 Concluding remarks

A general linear (or nonlinear) SDP

= "Optimization problem involving an $n \times n$ real symmetric matrix variable $\boldsymbol{X}$ to be positive semidefinite"

min.	a linear (or nonlinear) function in $\boldsymbol{y} \in \mathbb{R}^m$, $\boldsymbol{X} \in \mathbb{S}^n$,
sub. to	linear (or nonlinear) equalities and inequalies in $\boldsymbol{y} \in \mathbb{R}^m$, $\boldsymbol{X} \in \mathbb{S}^n$,

$$\boldsymbol{X} = \begin{pmatrix} X_{11} & X_{12} & \ldots & X_{1n} \\ X_{21} & X_{22} & \ldots & X_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ X_{n1} & X_{n2} & \ldots & X_{nn} \end{pmatrix} \succeq \boldsymbol{O}$$

(positive semidefinite).

Here $\mathbb{S}^n$ denotes the space of $n \times n$ symmetric matrices.

- We can solve linear SDP by interior-point methods.
- We will discuss 2 types of conversions of a large-scale SDP satisfying a structured sparsiy to solve it efficiently.

Applications of SDPs

- System and control theory — Linear matrix inequality
- Robust Optimization
- Machine learning
- Quantum chemistry
- Quantum computation
- Moment problems (Applied probablity)
- SDP relaxation —

  Max cut, Max clique, Sensor network localization,
  Polynomial optimization
- Design optimization of structures
- . . .

In many applications, SDPs are large-scale and often satisfy a certain sparsity characterized by a chordal graph structure.

Outline

0 Semidefinite Programming (SDP)

1 A simple example for 2 types of sparsities

2 Chordal graph

3 Domain-space sparsity

4 Range-space sparsity

5 Numerical results

6 Concluding remarks

Choose $b_i \in [0,1]$ $(i = 1, 2, \dots, n-1)$ randomly. A linear SDP:

$$\min \sum_{i=1}^{n-1} (X_{ii} + b_i(X_{i,i+1} + X_{i+1,i})) + X_{nn} \quad \text{— (1)}$$

sub. to (Matrix inequality, diagonal+bordered)

$$\boldsymbol{M}(\boldsymbol{X}) = \begin{pmatrix} 1 - X_{11} & 0 & \dots & X_{12} \\ 0 & 1 - X_{22} & \dots & X_{23} \\ \dots & \dots & \ddots & \dots \\ X_{21} & X_{32} & \dots & 1 - X_{nn} \end{pmatrix} \succeq \boldsymbol{O} \quad \text{— (2)}$$

$$\boldsymbol{X} = \begin{pmatrix} X_{11} & X_{12} & \dots & X_{1n} \\ X_{21} & X_{22} & \dots & X_{2n} \\ \dots & \dots & \dots & \dots \\ X_{n1} & X_{n2} & \dots & X_{nn} \end{pmatrix} \succeq \boldsymbol{O} \quad \text{(positive semidefinite)}$$

- The number of variables is $n(n+1)/2$; $X_{ij} = X_{ji}$.
- domain-space sparsity — Only X_{ij} $(|i-j| \leq 1)$ are used in (1), (2) among all variables X_{ij} $(1 \leq i \leq j \leq n)$.
- range-space sparsity — (2) is diagonal + bordered.

$\Downarrow$ Conversion with exploiting the domain and range sparsities

$$\min \sum_{i=1}^{n-1} (X_{ii} + b_i(X_{i,i+1} + X_{i+1,i})) + X_{nn} \quad \text{sub.to}$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} X_{11} & -X_{12} \\ -X_{12} & -z_1 \end{pmatrix} \succeq \boldsymbol{O},$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} X_{ii} & -X_{i,i+1} \\ -X_{i,i+1} & z_{i-1} - z_i \end{pmatrix} \succeq \boldsymbol{O} \ (i = 2, 3, \ldots, n-2),$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} X_{n-1,n-1} & -X_{n-1,n} \\ -X_{n-1,n} & X_{n,n} + z_{n-2} \end{pmatrix} \succeq \boldsymbol{O},$$

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} -X_{ii} & -X_{i,i+1} \\ -X_{i,i+1} & -X_{i+1,i+1} \end{pmatrix} \succeq \boldsymbol{O} \ (i = 1, 2, \ldots, n-1).$$

- The two SDPs are equivalent.
- $(3n-3)$ 2×2 linear matrix inequalities.
- $(3n-3)$ variables; the missing variables can be restored.

Numerical results

- SeDuMi (MATLAB, a prima-dual interior-point method)
- 2.66 GHz Dual-Core Intel Xeon with 12GB memory

	SeDuMi elapsed time (second)	
size of $\boldsymbol{X}$ $= n$	Original SDP	Converted SDP with exploiting d-space & r-space sparsities
10	0.2	0.1
100	1091.4	0.6
1000	-	6.3
10000	-	99.2

- Converted SDP satisfies another type of sparsity, the correlative sparsity, which makes the primal-dual interior-point method to work on it efficiently
— not discussed here.

Outline

0 Semidefinite Programming (SDP)

1 A simple example for 2 types of sparsities

2 Chordal graph

3 Domain-space sparsity

4 Range-space sparsity

5 Numerical results

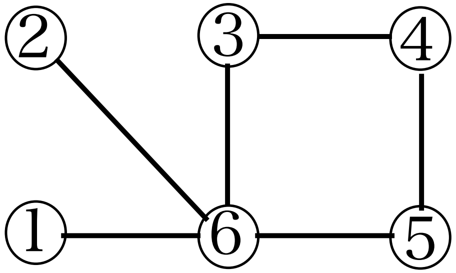
6 Concluding remarks

- Sparsity pattern will be described in terms of a graph.
- We will assume that the sparsity pattern graph has a sparse chordal extension to exploit the domain- and range-space sparsity in SDPs.

$G(N, E)$: a graph, $N = \{1, \ldots, n\}$ (nodes), $E \subset N \times N$ (edges)

chordal $\Leftrightarrow \forall$ cycle with more than 3 edges has a chord

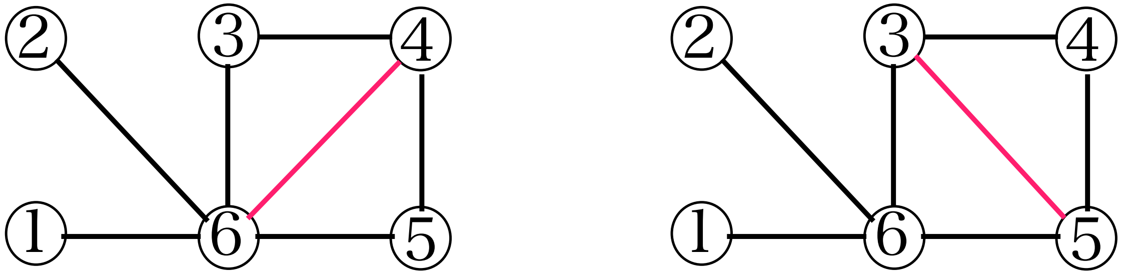
$G(N, E)$



Not chordal

$\Downarrow$ chordal extension

$G(N, \overline{E})$



(a) (b)

$\{1, 6\}, \{2, 6\}, \{3, 4, 6\}, \{4, 5, 6\}$

$\{1, 6\}, \{2, 6\}, \{3, 5, 6\}, \{3, 4, 5\}$

Maximal cliques (node sets of maximal complete subgraphs)

Outline

0 Semidefinite Programming (SDP)

1 A simple example for 2 types of sparsities

2 Chordal graph

3 Domain-space sparsity

4 Range-space sparsity

5 Numerical results

6 Concluding remarks

Opt. problem involving a symmetric matrix variable $\boldsymbol{X} \succeq \boldsymbol{O}$:

(P) min $f_0(\boldsymbol{y}, \boldsymbol{X})$ sub.to $\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{X}) \in \boldsymbol{\Omega}$, $\mathbb{S}^n \ni \boldsymbol{X} \succeq \boldsymbol{O}$.

Here $f_0 : \mathbb{R}^s \times \mathbb{S}^n \to \mathbb{R}$, $\boldsymbol{f} : \mathbb{R}^s \times \mathbb{S}^n \to V \supset \boldsymbol{\Omega}$.

d-space sparsity pattern graph $G(N, F)$: $N = \{1, 2, \ldots, n\}$,

$$F = \left\{ (i,j) : \begin{array}{l} i \neq j,\ X_{ij} \text{ is necessary} \\ \text{to evaluate } f_0(\boldsymbol{y}, \boldsymbol{X}) \text{ or } \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{X}) \end{array} \right\}$$

min $\quad f_0(\boldsymbol{y}, \boldsymbol{X}) = \sum_{i=1}^{3} (y_i X_{ii} + X_{i,i+1} + X_{i+1,i})$

sub. to

$$\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{X}) = \begin{pmatrix} 1 - X_{11} & X_{12} & y_1 & 2y_2 \\ X_{21} & 1 - X_{22} & X_{23} & 3y_3 \\ y_1 & X_{32} & 1 - X_{33} & X_{34} \\ 2y_2 & 3y_3 & X_{43} & 1 - X_{44} \end{pmatrix} \succeq \boldsymbol{O},$$

$\mathbb{S}^4 \ni \boldsymbol{X} \succeq \boldsymbol{O} \qquad \Rightarrow N = \{1, 2, 3, 4\}$

- X_{ij}, $|i - j| \leq 1$ are necessary to evaluate $f_0(\boldsymbol{y}, \boldsymbol{X})$, $\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{X})$
- $F = \{(i, i+1) : i = 1, 2, 3\}$

$G(N, F)$ = a chordal graph ① —— ② —— ③ —— ④

Opt. problem involving a symmetric matrix variable $\boldsymbol{X} \succeq \boldsymbol{O}$:

(P) min $f_0(\boldsymbol{y}, \boldsymbol{X})$ sub.to $\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{X}) \in \boldsymbol{\Omega}$, $\mathbb{S}^n \ni \boldsymbol{X} \succeq \boldsymbol{O}$.

Here $f_0 : \mathbb{R}^s \times \mathbb{S}^n \to \mathbb{R}$, $\boldsymbol{f} : \mathbb{R}^s \times \mathbb{S}^n \to V \supset \boldsymbol{\Omega}$.

d-space sparsity pattern graph $G(N, F)$: $N = \{1, 2, \ldots, n\}$,

$$F = \left\{ (i,j) : \begin{array}{l} i \neq j,\ X_{ij} \text{ is necessary} \\ \text{to evaluate } f_0(\boldsymbol{y}, \boldsymbol{X}) \text{ or } \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{X}) \end{array} \right\}$$

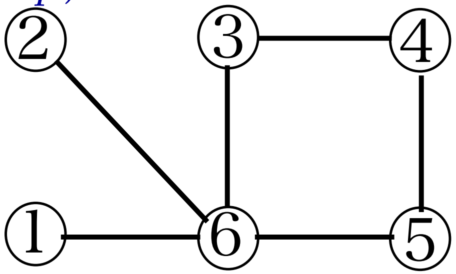
$\Updownarrow$

$G(N, E)$: a chordal extension of $G(N, F)$
$C_1, C_2, \ldots, C_\ell$: the maximal cliques of $G(N, E)$

(P') min $f_0(\boldsymbol{y}, \boldsymbol{X})$ sub.to $\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{X}) \in \boldsymbol{\Omega}$, $\boldsymbol{X}(C_p) \succeq \boldsymbol{O}$ $(p = 1, \ldots, \ell)$.

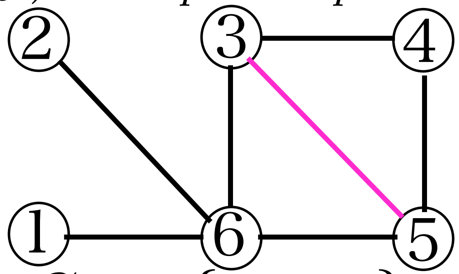
Here $\boldsymbol{X}(C_p)$: a submatrix consisting of X_{ij}, $(i, j) \in C_p \times C_p$.

$G(N, F)$



c. ext. $\Rightarrow$

$G(N, E)$
$C_1 = \{1, 6\}$
$C_2 = \{2, 6\}$
$C_3 = \{3, 4, 5\}$, $C_4 = \{3, 5, 6\}$.



$$\boldsymbol{X}(C_1) = \begin{pmatrix} X_{11} & X_{16} \\ X_{61} & X_{66} \end{pmatrix}, \boldsymbol{X}(C_2), \boldsymbol{X}(C_3), \boldsymbol{X}(C_4) \succeq \boldsymbol{O}.$$

Opt. problem involving a symmetric matrix variable $\boldsymbol{X} \succeq \boldsymbol{O}$:

(P) min $f_0(\boldsymbol{y}, \boldsymbol{X})$ sub.to $\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{X}) \in \boldsymbol{\Omega}$, $\mathbb{S}^n \ni \boldsymbol{X} \succeq \boldsymbol{O}$.

Here $f_0 : \mathbb{R}^s \times \mathbb{S}^n \rightarrow \mathbb{R}$, $\boldsymbol{f} : \mathbb{R}^s \times \mathbb{S}^n \rightarrow V \supset \boldsymbol{\Omega}$.

d-space sparsity pattern graph $G(N, F)$: $N = \{1, 2, \ldots, n\}$,

$$F = \left\{ (i, j) : \begin{array}{l} i \neq j,\ X_{ij} \text{ is necessary} \\ \text{to evaluate } f_0(\boldsymbol{y}, \boldsymbol{X}) \text{ or } \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{X}) \end{array} \right\}$$

$\Updownarrow$

$G(N, E)$: a chordal extension of $G(N, F)$
$C_1, C_2, \ldots, C_\ell$: the maximal cliques of $G(N, E)$

(P') min $f_0(\boldsymbol{y}, \boldsymbol{X})$ sub.to $\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{X}) \in \boldsymbol{\Omega}$, $\boldsymbol{X}(C_p) \succeq \boldsymbol{O}$ $(p = 1, \ldots, \ell)$.

Here $\boldsymbol{X}(C_p)$: a submatrix consisting of X_{ij}, $(i, j) \in C_p \times C_p$.

- (P) $\Leftrightarrow$ (P') is based on the positive definite matrix completion (Grone et al. 1984).

Outline

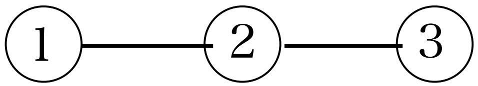
$G(N,E)$: a chordal graph with $N = \{1,\ldots,n\}$ and the max. cliques of $C_1,\ldots,C_\ell$. $E^\bullet = E \cup \{(i,i) : i \in N\}$.

$\mathbb{S}^n(E^\bullet) = \{\boldsymbol{Y} \in \mathbb{S}^n : Y_{ij} = 0 \ (i,j) \notin E^\bullet\}$.

$\mathbb{S}^C_+ = \{\boldsymbol{Y} \succeq \boldsymbol{O} : Y_{ij} = 0$ if $(i,j) \notin C \times C\}$ for $\forall C \subseteq N$.

Theorem (Agler et al. 1988)
Suppose $\boldsymbol{M} : \mathbb{R}^m \to \mathbb{S}^n(E^\bullet)$. $\boldsymbol{M}(\boldsymbol{u}) \succeq \boldsymbol{O}$ iff

$$\boldsymbol{M}(\boldsymbol{u}) = \boldsymbol{Y}^1 + \boldsymbol{Y}^2 + \cdots + \boldsymbol{Y}^\ell \text{ for } \exists \boldsymbol{Y}^k \in \mathbb{S}^{C_k}_+ \ (k = 1,\ldots,\ell).$$



$C_1 = \{1,2\}$, $C_2 = \{2,3\}$. $\boldsymbol{M} : \mathbb{R}^m \to \mathbb{S}^3(E^\bullet)$.

$$\boldsymbol{M}(\boldsymbol{u}) = \begin{pmatrix} M_{11}(\boldsymbol{u}) & M_{12}(\boldsymbol{u}) & 0 \\ M_{21}(\boldsymbol{u}) & M_{22}(\boldsymbol{u}) & M_{23}(\boldsymbol{u}) \\ 0 & M_{32}(\boldsymbol{u}) & M_{33}(\boldsymbol{u}) \end{pmatrix}$$

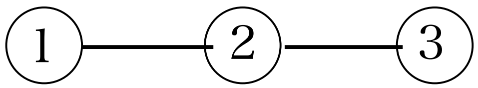
$G(N,E)$: a chordal graph with $N = \{1,\ldots,n\}$ and the max. cliques of $C_1,\ldots,C_\ell$. $E^\bullet = E \cup \{(i,i) : i \in N\}$.

$\mathbb{S}^n(E^\bullet) = \{\boldsymbol{Y} \in \mathbb{S}^n : Y_{ij} = 0 \ (i,j) \notin E^\bullet\}$.

$\mathbb{S}^C_+ = \{\boldsymbol{Y} \succeq \boldsymbol{O} : Y_{ij} = 0$ if $(i,j) \notin C \times C\}$ for $\forall C \subseteq N$.

Theorem (Agler et al. 1988)
Suppose $\boldsymbol{M} : \mathbb{R}^m \to \mathbb{S}^n(E^\bullet)$. $\boldsymbol{M}(\boldsymbol{u}) \succeq \boldsymbol{O}$ iff

$\boldsymbol{M}(\boldsymbol{u}) = \boldsymbol{Y}^1 + \boldsymbol{Y}^2 + \cdots + \boldsymbol{Y}^\ell$ for $\exists \boldsymbol{Y}^k \in \mathbb{S}^{C_k}_+ \ (k = 1,\ldots,\ell)$.



$C_1 = \{1,2\}$, $C_2 = \{2,3\}$. $\boldsymbol{M} : \mathbb{R}^m \to \mathbb{S}^3(E^\bullet)$.

$$\boldsymbol{M}(\boldsymbol{u}) = \begin{pmatrix} Y^1_{11} & Y^1_{12} & 0 \\ Y^1_{12} & Y^1_{22} & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & Y^2_{22} & Y^2_{23} \\ 0 & Y^2_{32} & Y^2_{33} \end{pmatrix}$$

$\boldsymbol{M}(\boldsymbol{u}) \succeq \boldsymbol{O}$

$\Updownarrow$

$$\left.\begin{array}{l} M_{11} = Y^1_{11}, M_{12} = Y^1_{12}, \\ M_{22} = Y^1_{22} + Y^2_{22}, \\ M_{23} = Y^2_{23}, M_{33} = Y^2_{33}, \\ \square \succeq \boldsymbol{O}, \ \square \succeq \boldsymbol{O} \end{array}\right\} \Leftrightarrow \left\{\begin{array}{l} \begin{pmatrix} M_{11}(\boldsymbol{u}) & M_{12}(\boldsymbol{u}) \\ M_{21}(\boldsymbol{u}) & Y^1_{22} \end{pmatrix} \succeq \boldsymbol{O}, \\ \begin{pmatrix} M_{22}(\boldsymbol{u}) - Y^1_{22} & M_{23}(\boldsymbol{u}) \\ M_{32}(\boldsymbol{u}) & M_{33}(\boldsymbol{u}) \end{pmatrix} \succeq \boldsymbol{O} \end{array}\right.$$

Summary of the d-space and r-space conversion methods:

Sparsity characterized by a chordal graph structure

$\Downarrow$

SDP (linear, polynomial, nonlinear)

each large-scale matrix variable

$\Downarrow$ exploiting d-space sparsity

multiple smaller matrix variables

each large-scale matrix inequality

$\Downarrow$ exploiting r-space sparsity

multiple smaller matrix inequalities

$\rightarrow$ SparseCoLO for linear SDP

$\Downarrow$ if SDP is linear $\qquad$ $\Downarrow$ relaxation if SDP is polynomial

Linear SDP with multiple smaller matrix variables and matrix inequalities

- SparsePOP = sparse SDP relaxation (Waki et. al ’06) :

POP $\Rightarrow$ Poly. SDP $\Rightarrow$ Linear SDP

adding valid poly. mat. inequalities $\leftarrow$ sparsity $\nearrow$ relaxation

Outline

0 Semidefinite Programming (SDP)

1 A simple example for 2 types of sparsities

2 Chordal graph

3 Domain-space sparsity

4 Range-space sparsity

5 Numerical results

6 Concluding remarks

Test Problems

(a) Linear SDP relaxation of randomly generated sparse quadratic SDPs

(b) SDP relaxation of quadratic optimization problems (QOPs)

(c) Polynomial optimization problems (POPs)

- We apply SparseCoLO+ SDPA to (a) and (b), where SparseCoLO — MATLAB software for the d-space and r-space conversion methods,
  SDPA — a primal-dual interior-point method for SDPs.
- We apply SparsePOP + SDPA to (c), where SparsePOP — a sparse SDP relaxation for POPs using the d-space conversion method.

- 3.06 GHz Intel Core 2 Duo with 8 GB memory.

(a) Linear SDP relaxation of a sparse quadratic SDP

Quadratic SDP: min $\boldsymbol{c}^T\boldsymbol{x}$ sub to $\boldsymbol{M}(\boldsymbol{x}) \succeq \boldsymbol{O}$,

where $\boldsymbol{M} : \mathbb{R}^s \rightarrow \mathbb{S}^n$ whose (i,j) element is given by

$$M_{ij}(\boldsymbol{x}) = (1, \boldsymbol{x}^T)\boldsymbol{Q}_{ij}\begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix} = \boldsymbol{Q}_{ij} \bullet \begin{pmatrix} 1 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{x}\boldsymbol{x}^T \end{pmatrix}, \ \forall \boldsymbol{x} \in \mathbb{R}^s.$$

Here $\boldsymbol{Q} \bullet \boldsymbol{Y} =$ trace $\boldsymbol{Q}^T\boldsymbol{Y}$ (the inner product of $\boldsymbol{Q}$ and $\boldsymbol{Y}$).

(a) Linear SDP relaxation of a sparse quadratic SDP

SDP: min $\boldsymbol{c}^T\boldsymbol{x}$ sub to $\widehat{\boldsymbol{M}}(\boldsymbol{x},\boldsymbol{X}) \succeq \boldsymbol{O}$, $\begin{pmatrix} x_0 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{pmatrix} \succeq \boldsymbol{O}$, $x_0 = 1$,

where $\widehat{\boldsymbol{M}} : \mathbb{R}^s \times \mathbb{S}^s \rightarrow \mathbb{S}^n$ whose (i,j) element is given by

$\widehat{M}_{ij}(\boldsymbol{x},\boldsymbol{X}) = \boldsymbol{Q}_{ij} \bullet \begin{pmatrix} 1 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{pmatrix}$ for every $\boldsymbol{x} \in \mathbb{R}^s$, $\boldsymbol{X} \in \mathbb{S}^s$,

$\Uparrow$ Linear SDP relaxation

Quadratic SDP: min $\boldsymbol{c}^T\boldsymbol{x}$ sub to $\boldsymbol{M}(\boldsymbol{x}) \succeq \boldsymbol{O}$,

where $\boldsymbol{M} : \mathbb{R}^s \rightarrow \mathbb{S}^n$ whose (i,j) element is given by

$$M_{ij}(\boldsymbol{x}) = (1, \boldsymbol{x}^T)\boldsymbol{Q}_{ij} \begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix} = \boldsymbol{Q}_{ij} \bullet \begin{pmatrix} 1 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{x}\boldsymbol{x}^T \end{pmatrix}, \ \forall \boldsymbol{x} \in \mathbb{R}^s.$$

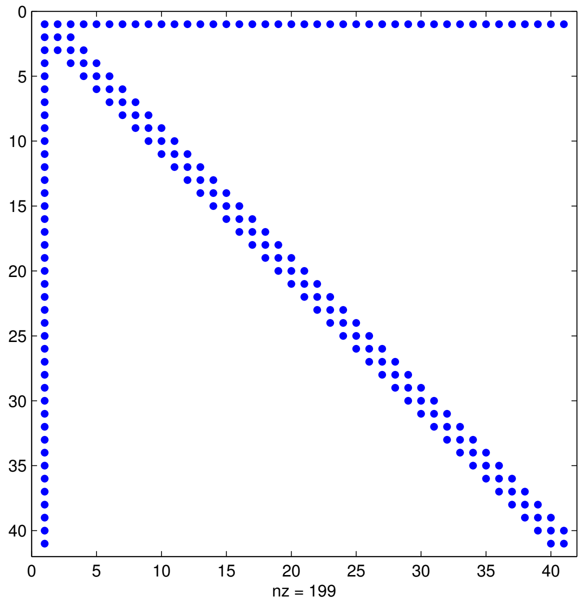
Here $\boldsymbol{Q} \bullet \boldsymbol{Y} =$ trace $\boldsymbol{Q}^T\boldsymbol{Y}$ (the inner product of $\boldsymbol{Q}$ and $\boldsymbol{Y}$).

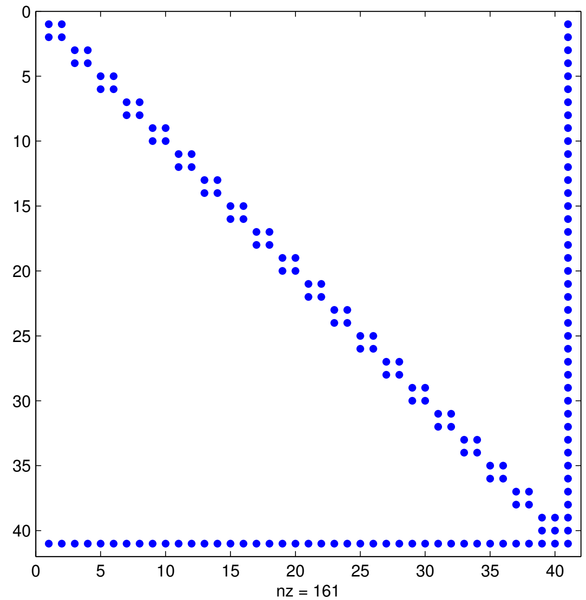
(a) Linear SDP relaxation of a sparse quadratic SDP

SDP: min $\boldsymbol{c}^T\boldsymbol{x}$ sub to $\widehat{\boldsymbol{M}}(\boldsymbol{x},\boldsymbol{X}) \succeq \boldsymbol{O}$, $\begin{pmatrix} x_0 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{pmatrix} \succeq \boldsymbol{O}$, $x_0 = 1$,

where $\widehat{\boldsymbol{M}} : \mathbb{R}^s \times \mathbb{S}^s \to \mathbb{S}^n$ whose (i,j) element is given by

$$\widehat{M}_{ij}(\boldsymbol{x},\boldsymbol{X}) = \boldsymbol{Q}_{ij} \bullet \begin{pmatrix} 1 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{pmatrix}$$ for every $\boldsymbol{x} \in \mathbb{R}^s$, $\boldsymbol{X} \in \mathbb{S}^s$,





d-space sparsity ($\forall \boldsymbol{Q}_{ij}$) and r–space sparsity ($\widehat{\boldsymbol{M}}$)

$(s = 40,\ n = 41)$

(a) Linear SDP relaxation of a sparse quadratic SDP

SDP: min $\boldsymbol{c}^T\boldsymbol{x}$ sub to $\widehat{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{X}) \succeq \boldsymbol{O}$, $\begin{pmatrix} x_0 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{pmatrix} \succeq \boldsymbol{O}$, $x_0 = 1$,

where $\widehat{\boldsymbol{M}} : \mathbb{R}^s \times \mathbb{S}^s \rightarrow \mathbb{S}^n$ whose (i, j) element is given by

$$\widehat{M}_{ij}(\boldsymbol{x}, \boldsymbol{X}) = \boldsymbol{Q}_{ij} \bullet \begin{pmatrix} 1 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{pmatrix} \text{ for every } \boldsymbol{x} \in \mathbb{R}^s, \boldsymbol{X} \in \mathbb{S}^s,$$

		SDPA elapsed time in seconds			
s	n	no sparsity	d-space	r-space	d- & r-space
40	41	1.4	0.3	1.3	0.2
80	81	33.5	1.7	34.6	0.8
160	161	1427.1	19.6	1483.0	4.1
320	321	-	262.2	-	31.8

(b) Linear SDP relaxation of sparse QOPs

Sparse Linear SDP	size X	No. of equalities	E. time in seconds: no sparsity	E. time in seconds: d-space
M1000.05	1000	1000	41.2	0.5
M1000.15	1000	1000	39.6	52.7
thetaG11	801	2401	41.8	6.9
qpG11	1600	800	112.5	3.1
sensor1000	1002	11010	271.8	18.3
sensor4000	4002	47010	o.mem.	56.0

Sparse Linear SDP — sparse QOP

M1000.?? $\Leftarrow$ max cut problems with diff. edge densities

thetaG11 $\Leftarrow$ minimization of the Lovasz theta function

qpG11 $\Leftarrow$ a box constrained QOP

sensor???? $\Leftarrow$ a sensor network localization problem with ???? sensors

(c) SDP relaxation of POPs by SparsePOP+SDPA — 1

alkyl from globalib

$$
\begin{array}{ll}
\min & -6.3x_5x_8 + 5.04x_2 + 0.35x_3 + x_4 + 3.36x_6 \\
\text{sub.to} & -0.820x_2 + x_5 - 0.820x_6 = 0, \\
& 0.98x_4 - x_7(0.01x_5x_{10} + x_4) = 0, \\
& -x_2x_9 + 10x_3 + x_6 = 0, \\
& x_5x_{12} - x_2(1.12 + 0.132x_9 - 0.0067x_9^2) = 0, \\
& x_8x_{13} - 0.01x_9(1.098 - 0.038x_9) - 0.325x_7 = 0.574, \\
& x_{10}x_{14} + 22.2x_{11} = 35.82, \\
& x_1x_{11} - 3x_8 = -1.33,\ \text{lbd}_i \leq x_i \leq \text{ubd}_i \ (i = 1, 2, \ldots, 14).
\end{array}
$$

no sparsity	d-space eparsity		
E. time	E. time	ϵ_{obj}	ϵ_{feas}
> 10,000	1.3	8.2e-6	8.5e-10

ϵ_{obj} = approx. min. val. - lower bd. for the min. val.,

ϵ_{feas} = the max. error in equalities.

(c) SDP relaxation of POPs by SparsePOP+SDPA — 2

Minimize the Broyden tridiagonal function $f_B(\boldsymbol{x})$ over $\mathbb{R}^n$.

$$f_B(\boldsymbol{x}) \quad = \quad \sum_{i=1}^{n} ((3-2x_i)x_i - x_{i-1} - 2x_{i+1} + 1)^2 ,$$

where $x_0 = 0$ and $x_{n+1} = 0$.

	no sparsity	d-space	
n	E. time	E. time	ϵ_{obj}
10	1.80	0.04	4.4e-9
20	916.95	0.08	1.5e-9
5000	o.mem.	29.44	5.1e-5
10000	o.mem.	59.52	9.2e-4

ϵ_{obj} $=$ an approx. min. val. - a l. bound for the min. val..

Outline

0 Semidefinite Programming (SDP)

1 A simple example for 2 types of sparsities

2 Chordal graph

3 Domain-space sparsity

4 Range-space sparsity

5 Numerical results

6 Concluding remarks

Two types of sparsities of large-scale SDPs which are characterized by a chordal graph structure:

(a) Domain-space sparsity
(b) Range-space sparsity

- Numerical methods for converting large-scale SDPs into smaller SDPs by exploiting (a) and (b).

Linear,	each large-scale matrix variable
polynomial or	$\Downarrow$ exploiting (a) Domain-space sparsity
nonlinear	multiple smaller matrix variables
SDP	each large-scale matrix inequality
	$\Downarrow$ exploiting (b) Range-space sparsity
	multiple smaller matrix inequalities

- Very effective when SDP is sparse.
- Overheads in domain- & range-space conversion methods; adding equalities, real variables and/or matrix variables. Hence, less effective if SDP is denser.