Accelerated first-order methods for a class of semidefinite programs

Alex L. Wang, CMU $\rightarrow$ CWI $\rightarrow$ Purdue University
Fatma Kılınç-Karzan, Carnegie Mellon University

SIAM OP23, June, 2023

1. What structure? . . . k-exactness
2. How to use k-exactness? . . . Strongly convex reformulation
3. Algorithms
4. Conclusion

Standard SDP algorithms are expensive

- Semidefinite program

Standard SDP algorithms are expensive

- Semidefinite program

$$(\textbf{SDP}) = \inf_{Y \in \mathbb{S}^{n+k}} \left\{ \langle M_0, Y \rangle : \begin{array}{l} \langle M_i, Y \rangle + d_i = 0,\ \forall i \in [m] \\ Y \succeq 0 \end{array} \right\}$$

Standard SDP algorithms are expensive

- Semidefinite program

$$(\textbf{SDP}) = \inf_{Y \in \mathbb{S}^{n+k}} \left\{ \langle M_0, Y \rangle : \begin{array}{l} \langle M_i, Y \rangle + d_i = 0,\ \forall i \in [m] \\ Y \succeq 0 \end{array} \right\}$$

$$\geq \sup_{\gamma \in \mathbb{R}^m} \left\{ d^\mathsf{T} \gamma : \ M_0 + \sum_{i=1}^{m} \gamma_i M_i \succeq 0 \right\}$$

Standard SDP algorithms are expensive

- Semidefinite program

$$(\textbf{SDP}) = \inf_{Y \in \mathbb{S}^{n+k}} \left\{ \langle M_0, Y \rangle : \begin{array}{l} \langle M_i, Y \rangle + d_i = 0,\ \forall i \in [m] \\ Y \succeq 0 \end{array} \right\}$$

$$\geq \sup_{\gamma \in \mathbb{R}^m} \left\{ d^\intercal \gamma : \underbrace{M_0 + \sum_{i=1}^{m} \gamma_i M_i}_{M(\gamma)} \succeq 0 \right\}$$

Standard SDP algorithms are expensive

- Semidefinite program

$$(\textbf{SDP}) = \inf_{Y \in \mathbb{S}^{n+k}} \left\{ \langle M_0, Y \rangle : \begin{array}{l} \langle M_i, Y \rangle + d_i = 0, \ \forall i \in [m] \\ Y \succeq 0 \end{array} \right\}$$

$$\geq \sup_{\gamma \in \mathbb{R}^m} \left\{ d^\intercal \gamma : \underbrace{M_0 + \sum_{i=1}^m \gamma_i M_i}_{M(\gamma)} \succeq 0 \right\}$$

  - Convex and expressive, but most existing methods too slow for large $n + k$

Standard SDP algorithms are expensive

- Semidefinite program

$$(\textbf{SDP}) = \inf_{Y \in \mathbb{S}^{n+k}} \left\{ \langle M_0, Y \rangle : \begin{array}{l} \langle M_i, Y \rangle + d_i = 0,\ \forall i \in [m] \\ Y \succeq 0 \end{array} \right\}$$

$$\geq \sup_{\gamma \in \mathbb{R}^m} \left\{ d^\intercal \gamma : \underbrace{M_0 + \sum_{i=1}^m \gamma_i M_i}_{M(\gamma)} \succeq 0 \right\}$$

  - Convex and expressive, but most existing methods too slow for large $n + k$
  - Simply writing down Y requires $O((n+k)^2)$ memory

QMPs and their SDP relaxations

- Quadratic matrix program

Related: Beck [2007], Beck et al. [2012], Wang and Kılınç-Karzan [2020]

QMPs and their SDP relaxations

- Quadratic matrix program

$$(\textbf{QMP}) = \inf_{X \in \mathbb{R}^{n \times k}} \{q_0(X) : q_i(X) = 0, \forall i \in [m]\}$$

Related: Beck [2007], Beck et al. [2012], Wang and Kılınç-Karzan [2020]

QMPs and their SDP relaxations

- Quadratic matrix program

$$(\textbf{QMP}) = \inf_{X\in\mathbb{R}^{n\times k}} \{q_0(X) : q_i(X) = 0, \forall i \in [m]\}$$

where $q_i(X) = \operatorname{tr}(X^\intercal A_i X) + 2\langle B_i, X\rangle + c_i$ is a quadratic matrix function

Related: Beck [2007], Beck et al. [2012], Wang and Kılınç-Karzan [2020]

QMPs and their SDP relaxations

- Quadratic matrix program

$$(\textbf{QMP}) = \inf_{X\in\mathbb{R}^{n\times k}} \{q_0(X) : q_i(X) = 0,\ \forall i \in [m]\}$$

where $q_i(X) = \operatorname{tr}(X^\intercal A_i X) + 2\langle B_i, X\rangle + c_i$ is a quadratic matrix function

- SDP relaxation of QMP

Related: Beck [2007], Beck et al. [2012], Wang and Kılınç-Karzan [2020]

QMPs and their SDP relaxations

- Quadratic matrix program

$$(\textbf{QMP}) = \inf_{X \in \mathbb{R}^{n\times k}} \{q_0(X) : q_i(X) = 0, \ \forall i \in [m]\}$$

where $q_i(X) = \operatorname{tr}(X^\intercal A_i X) + 2\langle B_i, X\rangle + c_i$ is a quadratic matrix function

- SDP relaxation of QMP

$$q_i(X) = \left\langle \begin{pmatrix} A_i & B_i \\ B_i^\intercal & c_i I_k/k \end{pmatrix}, \begin{pmatrix} XX^\intercal & X \\ X^\intercal & I_k \end{pmatrix} \right\rangle = \left\langle M_i, \begin{pmatrix} XX^\intercal & X \\ X^\intercal & I_k \end{pmatrix} \right\rangle$$

Related: Beck [2007], Beck et al. [2012], Wang and Kılınç-Karzan [2020]

QMPs and their SDP relaxations

- Quadratic matrix program

$$(\textbf{QMP}) = \inf_{X\in\mathbb{R}^{n\times k}} \{q_0(X) : q_i(X) = 0,\ \forall i \in [m]\}$$

where $q_i(X) = \operatorname{tr}(X^\intercal A_i X) + 2\langle B_i, X\rangle + c_i$ is a quadratic matrix function

- SDP relaxation of QMP

$$q_i(X) = \left\langle \begin{pmatrix} A_i & B_i \\ B_i^\intercal & c_i I_k/k \end{pmatrix}, \begin{pmatrix} XX^\intercal & X \\ X^\intercal & I_k \end{pmatrix} \right\rangle = \left\langle M_i, \begin{pmatrix} XX^\intercal & X \\ X^\intercal & I_k \end{pmatrix} \right\rangle$$

$$(\textbf{QMP}) \geq \inf_{Y\in\mathbb{S}^{n+k}} \left\{ \langle M_0, Y\rangle : \begin{array}{l} \langle M_i, Y\rangle = 0,\ \forall i \in [m] \\ Y = \begin{pmatrix} * & * \\ * & I_k \end{pmatrix} \succeq 0 \end{array} \right\} = (\textbf{SDP})$$

Related: Beck [2007], Beck et al. [2012], Wang and Kılınç-Karzan [2020]

Structural assumptions on SDP

- $$(\textbf{SDP}) = \inf_{Y \in \mathbb{S}^{n+k}} \left\{ \langle M_0, Y \rangle : \begin{array}{l} \langle M_i, Y \rangle = 0,\ \forall i \in [m] \\ Y = \begin{pmatrix} * & * \\ * & I_k \end{pmatrix} \succeq 0 \end{array} \right\}$$

Related: Alizadeh et al. [1997], Ding et al. [2021]

Structural assumptions on SDP

- $$(\textbf{SDP}) = \inf_{Y \in \mathbb{S}^{n+k}} \left\{ \langle M_0, Y \rangle : \begin{array}{l} \langle M_i, Y \rangle = 0,\ \forall i \in [m] \\ Y = \begin{pmatrix} * & * \\ * & I_k \end{pmatrix} \succeq 0 \end{array} \right\}$$
- k**-exact SDPs**:

Related: Alizadeh et al. [1997], Ding et al. [2021]

Structural assumptions on SDP

- $$(\textbf{SDP}) = \inf_{Y \in \mathbb{S}^{n+k}} \left\{ \langle M_0, Y \rangle : \begin{array}{l} \langle M_i, Y \rangle = 0,\ \forall i \in [m] \\ Y = \begin{pmatrix} * & * \\ * & I_k \end{pmatrix} \succeq 0 \end{array} \right\}$$
- k**-exact SDPs**:
  - **Strong duality** holds, primal and dual are both solvable

Related: Alizadeh et al. [1997], Ding et al. [2021]

Structural assumptions on SDP

- $$(\textbf{SDP}) = \inf_{Y \in \mathbb{S}^{n+k}} \left\{ \langle M_0, Y \rangle : \begin{array}{l} \langle M_i, Y \rangle = 0, \ \forall i \in [m] \\ Y = \begin{pmatrix} * & * \\ * & I_k \end{pmatrix} \succeq 0 \end{array} \right\}$$
- k**-exact SDPs**:
  - **Strong duality** holds, primal and dual are both solvable
  - **Exactness**: (**QMP**) $=$ (**SDP**), i.e., there exists Y^* with $\mathrm{rank}(Y^*) = k$

Related: Alizadeh et al. [1997], Ding et al. [2021]

Structural assumptions on SDP

- $$(\textbf{SDP}) = \inf_{Y \in \mathbb{S}^{n+k}} \left\{ \langle M_0, Y \rangle : \begin{array}{l} \langle M_i, Y \rangle = 0, \ \forall i \in [m] \\ Y = \begin{pmatrix} * & * \\ * & I_k \end{pmatrix} \succeq 0 \end{array} \right\}$$
- k**-exact SDPs**:
  - **Strong duality** holds, primal and dual are both solvable
  - **Exactness**: (**QMP**) $=$ (**SDP**), i.e., there exists Y^* with $\mathrm{rank}(Y^*) = k$
  - **Strict complementarity**: there exists γ^* with $\mathrm{rank}(M(\gamma^*)) = n$

Related: Alizadeh et al. [1997], Ding et al. [2021]

Structural assumptions on SDP

- $$(\textbf{SDP}) = \inf_{Y \in \mathbb{S}^{n+k}} \left\{ \langle M_0, Y \rangle : \begin{array}{l} \langle M_i, Y \rangle = 0,\ \forall i \in [m] \\ Y = \begin{pmatrix} * & * \\ * & I_k \end{pmatrix} \succeq 0 \end{array} \right\}$$
- k**-exact SDPs**:
  - **Strong duality** holds, primal and dual are both solvable
  - **Exactness**: (**QMP**) $=$ (**SDP**), i.e., there exists Y^* with $\mathrm{rank}(Y^*) = k$
  - **Strict complementarity**: there exists γ^* with $\mathrm{rank}(M(\gamma^*)) = n$
- Have reduced SDP $\longrightarrow$ QMP:

$$Y^* = \begin{pmatrix} X^*(X^*)^\intercal & X^* \\ (X^*)^\intercal & I_k \end{pmatrix}$$

Related: Alizadeh et al. [1997], Ding et al. [2021]

1. What structure? . . . k-exactness
2. How to use k-exactness? . . . Strongly convex reformulation
3. Algorithms
4. Conclusion

Lagrangian reformulation

- Take the Lagrangian to get minimax problem without constraints

$$X^* = \arg\min_{X \in \mathbb{R}^{n\times k}} \{q_0(X) : \, q_i(X) = 0, \, \forall i \in [m]\}$$

Lagrangian reformulation

- Take the Lagrangian to get minimax problem without constraints

$$
\begin{aligned}
X^* &= \arg\min_{X \in \mathbb{R}^{n\times k}} \{q_0(X) : \, q_i(X) = 0, \, \forall i \in [m]\} \\
&= \arg\min_{X \in \mathbb{R}^{n\times k}} \sup_{\gamma \in \mathbb{R}^m} \; q_0(X) + \sum_{i=1}^m \gamma_i q_i(X)
\end{aligned}
$$

Lagrangian reformulation

- Take the Lagrangian to get minimax problem without constraints

$$X^* = \arg\min_{X \in \mathbb{R}^{n\times k}} \{q_0(X) : q_i(X) = 0,\ \forall i \in [m]\}$$

$$= \arg\min_{X \in \mathbb{R}^{n\times k}} \sup_{\gamma \in \mathbb{R}^m} \underbrace{q_0(X) + \sum_{i=1}^{m} \gamma_i q_i(X)}_{=:\ q(\gamma, X)}$$

Lagrangian reformulation

- Take the Lagrangian to get minimax problem without constraints

$$X^* = \operatorname*{arg\,min}_{X \in \mathbb{R}^{n \times k}} \{q_0(X) : q_i(X) = 0,\ \forall i \in [m]\}$$
$$= \operatorname*{arg\,min}_{X \in \mathbb{R}^{n \times k}} \sup_{\gamma \in \mathbb{R}^m} \underbrace{q_0(X) + \sum_{i=1}^m \gamma_i q_i(X)}_{=:\ q(\gamma, X)}$$

- $q(\gamma, X) = \operatorname{tr}(X^\intercal A(\gamma) X) + \dots$

Lagrangian reformulation

- Take the Lagrangian to get minimax problem without constraints

$$X^* = \operatorname*{arg\,min}_{X \in \mathbb{R}^{n\times k}} \{q_0(X) : q_i(X) = 0,\ \forall i \in [m]\}$$

$$= \operatorname*{arg\,min}_{X \in \mathbb{R}^{n\times k}} \sup_{\gamma \in \mathbb{R}^m} \underbrace{q_0(X) + \sum_{i=1}^m \gamma_i q_i(X)}_{=:\ q(\gamma, X)}$$

- $q(\gamma, X) = \operatorname{tr}(X^\intercal A(\gamma) X) + \ldots$ where

$$A(\gamma) = \text{top-left block of } M(\gamma)$$

Deriving a strongly convex minimax problem

- **Thought experiment**: strong duality + strict complementarity implies

$$X^* = \mathop{\arg\min}_{X \in \mathbb{R}^{n \times k}} \sup_{\gamma \in \mathbb{R}^m} q(\gamma, X)$$

$$= \mathop{\arg\min}_{X \in \mathbb{R}^{n \times k}} q(\gamma^*, X)$$

Deriving a strongly convex minimax problem

- **Thought experiment**: strong duality + strict complementarity implies

$$X^* = \mathop{\arg\min}_{X \in \mathbb{R}^{n\times k}} \sup_{\gamma \in \mathbb{R}^m} q(\gamma, X)$$
$$= \mathop{\arg\min}_{X \in \mathbb{R}^{n\times k}} q(\gamma^*, X)$$

- Strict complementarity

Deriving a strongly convex minimax problem

- **Thought experiment**: strong duality + strict complementarity implies

$$X^* = \operatorname*{arg\,min}_{X \in \mathbb{R}^{n \times k}} \sup_{\gamma \in \mathbb{R}^m} q(\gamma, X)$$

$$= \operatorname*{arg\,min}_{X \in \mathbb{R}^{n \times k}} q(\gamma^*, X)$$

- Strict complementarity $\longrightarrow$ $A(\gamma^*) \succ 0$

Deriving a strongly convex minimax problem

- **Thought experiment**: strong duality + strict complementarity implies

$$X^* = \arg\min_{X \in \mathbb{R}^{n\times k}} \sup_{\gamma \in \mathbb{R}^m} q(\gamma, X)$$

$$= \arg\min_{X \in \mathbb{R}^{n\times k}} q(\gamma^*, X)$$

- Strict complementarity $\longrightarrow A(\gamma^*) \succ 0 \longrightarrow q(\gamma^*, X)$ is strongly convex

Deriving a strongly convex minimax problem

- **Thought experiment**: strong duality + strict complementarity implies

$$X^* = \arg\min_{X\in\mathbb{R}^{n\times k}} \sup_{\gamma\in\mathbb{R}^m} q(\gamma, X)$$

$$= \arg\min_{X\in\mathbb{R}^{n\times k}} q(\gamma^*, X)$$

- Strict complementarity $\longrightarrow$ $A(\gamma^*) \succ 0$ $\longrightarrow$ $q(\gamma^*, X)$ is strongly convex

Theorem (Certificate of strict compl. gives strongly conv. reform.)

Suppose $\gamma^* \in \mathcal{C} \subseteq \mathbb{R}^m$ and $A(\gamma) \succ 0$ for all $\gamma \in \mathcal{C}$, then

$$X^* = \arg\min_{X\in\mathbb{R}^{n\times k}} \max_{\gamma\in\mathcal{C}} q(\gamma, X)$$

Deriving a strongly convex minimax problem

- **Thought experiment**: strong duality + strict complementarity implies

$$X^* = \arg\min_{X\in\mathbb{R}^{n\times k}} \sup_{\gamma\in\mathbb{R}^m} q(\gamma, X)$$

$$= \arg\min_{X\in\mathbb{R}^{n\times k}} q(\gamma^*, X)$$

- Strict complementarity $\longrightarrow A(\gamma^*) \succ 0 \longrightarrow q(\gamma^*, X)$ is strongly convex

Theorem (Certificate of strict compl. gives strongly conv. reform.)

Suppose $\gamma^* \in \mathcal{C} \subseteq \mathbb{R}^m$ and $A(\gamma) \succ 0$ for all $\gamma \in \mathcal{C}$, then

$$X^* = \arg\min_{X\in\mathbb{R}^{n\times k}} \max_{\gamma\in\mathcal{C}} q(\gamma, X)$$

- **Algorithm**:

Deriving a strongly convex minimax problem

- **Thought experiment**: strong duality + strict complementarity implies

$$X^* = \arg\min_{X \in \mathbb{R}^{n\times k}} \sup_{\gamma\in\mathbb{R}^m} q(\gamma, X)$$

$$= \arg\min_{X \in \mathbb{R}^{n\times k}} q(\gamma^*, X)$$

- Strict complementarity $\longrightarrow$ $A(\gamma^*) \succ 0$ $\longrightarrow$ $q(\gamma^*, X)$ is strongly convex

Theorem (Certificate of strict compl. gives strongly conv. reform.)

Suppose $\gamma^* \in \mathcal{C} \subseteq \mathbb{R}^m$ and $A(\gamma) \succ 0$ for all $\gamma \in \mathcal{C}$, then

$$X^* = \arg\min_{X \in \mathbb{R}^{n\times k}} \max_{\gamma\in\mathcal{C}} q(\gamma, X)$$

- **Algorithm**:
  - Construct $\mathcal{C}$

Deriving a strongly convex minimax problem

- **Thought experiment**: strong duality + strict complementarity implies

$$X^* = \arg\min_{X \in \mathbb{R}^{n\times k}} \sup_{\gamma \in \mathbb{R}^m} q(\gamma, X)$$

$$= \arg\min_{X \in \mathbb{R}^{n\times k}} q(\gamma^*, X)$$

- Strict complementarity $\longrightarrow A(\gamma^*) \succ 0 \longrightarrow q(\gamma^*, X)$ is strongly convex

Theorem (Certificate of strict compl. gives strongly conv. reform.)

Suppose $\gamma^* \in \mathcal{C} \subseteq \mathbb{R}^m$ and $A(\gamma) \succ 0$ for all $\gamma \in \mathcal{C}$, then

$$X^* = \arg\min_{X \in \mathbb{R}^{n\times k}} \max_{\gamma \in \mathcal{C}} q(\gamma, X)$$

- **Algorithm**:
  - Construct $\mathcal{C}$
  - Solve strongly convex quadratic matrix minimax problem (QMMP)

1. What structure? ... k-exactness
2. How to use k-exactness? ... Strongly convex reformulation
3. Algorithms
4. Conclusion

Algorithms: CautiousAGD

- How to solve strongly convex QMMP?

$$\arg\min_{X \in \mathbb{R}^{n\times k}} \max_{\gamma \in \mathcal{C}} q(\gamma, X)$$

Related: Nesterov [2005], Devolder et al. [2013, 2014], Nesterov [2018]

Algorithms: CautiousAGD

- How to solve strongly convex QMMP?

$$\arg\min_{X \in \mathbb{R}^{n\times k}} \max_{\gamma\in\mathcal{C}} q(\gamma, X)$$

  - Accelerated gradient descent (AGD) method for minimax functions

Related: Nesterov [2005], Devolder et al. [2013, 2014], Nesterov [2018]

- How to solve strongly convex QMMP?

$$\arg\min_{X\in\mathbb{R}^{n\times k}}\max_{\gamma\in\mathcal{C}} q(\gamma, X)$$

  - Accelerated gradient descent (AGD) method for minimax functions
  - **Issue**: Requires solving a “prox-map” in each iteration

Related: Nesterov [2005], Devolder et al. [2013, 2014], Nesterov [2018]

Algorithms: CautiousAGD

- How to solve strongly convex QMMP?

$$\arg\min_{X \in \mathbb{R}^{n \times k}} \max_{\gamma \in \mathcal{C}} q(\gamma, X)$$

  - Accelerated gradient descent (AGD) method for minimax functions
  - **Issue**: Requires solving a "prox-map" in each iteration
    - Solve prox-map approximately and bound error in AGD

Related: Nesterov [2005], Devolder et al. [2013, 2014], Nesterov [2018]

Algorithms: CautiousAGD

- How to solve strongly convex QMMP?

$$\arg\min_{X \in \mathbb{R}^{n \times k}} \max_{\gamma \in \mathcal{C}} q(\gamma, X)$$

  - Accelerated gradient descent (AGD) method for minimax functions
  - **Issue**: Requires solving a "prox-map" in each iteration
    - Solve prox-map approximately and bound error in AGD
  - $\longrightarrow$ CautiousAGD

Related: Nesterov [2005], Devolder et al. [2013, 2014], Nesterov [2018]

Algorithms: CautiousAGD

- How to solve strongly convex QMMP?

$$\arg\min_{X \in \mathbb{R}^{n\times k}} \max_{\gamma\in\mathcal{C}} q(\gamma, X)$$

  - Accelerated gradient descent (AGD) method for minimax functions
  - **Issue**: Requires solving a "prox-map" in each iteration
    - Solve prox-map approximately and bound error in AGD
  - $\longrightarrow$ CautiousAGD

Theorem (CautiousAGD)

CautiousAGD produces iterates X_t such that

$$\max_{\gamma\in\mathcal{C}} q(\gamma, X_t) \le \min_X \max_{\gamma\in\mathcal{C}} q(\gamma, X) + \epsilon$$

after $O\left(\log(\epsilon^{-1})\right)$ iterations, $O(m\epsilon^{-1/2})$ matrix-vector products per iteration

Related: Nesterov [2005], Devolder et al. [2013, 2014], Nesterov [2018]

Algorithms: CertSDP

- How to construct $\mathcal{C}$?

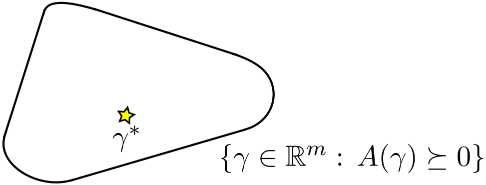
Algorithms: CertSDP

- How to construct $\mathcal{C}$? $\gamma^* \in \mathcal{C}$

Algorithms: CertSDP

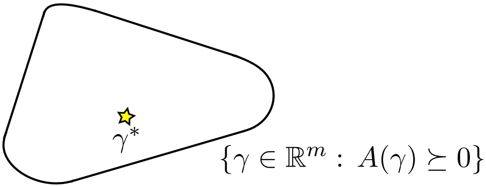
- How to construct $\mathcal{C}$? $\gamma^* \in \mathcal{C}$, $A(\gamma) \succ 0$ for all $\gamma \in \mathcal{C}$

- How to construct $\mathcal{C}$? $\gamma^* \in \mathcal{C}$, $A(\gamma) \succ 0$ for all $\gamma \in \mathcal{C}$



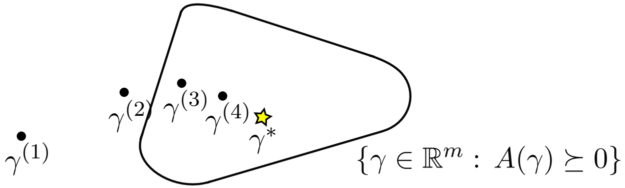
Algorithms: CertSDP

- How to construct $\mathcal{C}$? $\gamma^* \in \mathcal{C}$, $A(\gamma) \succ 0$ for all $\gamma \in \mathcal{C}$
  - γ^* is optimizer of dual problem, there exist (slow) algorithms $\gamma^{(i)} \to \gamma^*$

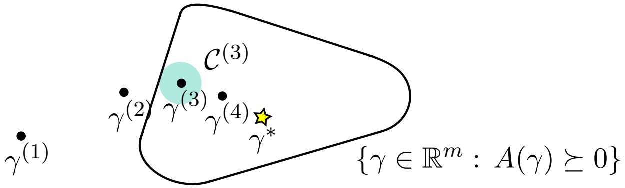


Algorithms: CertSDP

- How to construct $\mathcal{C}$? $\gamma^* \in \mathcal{C}$, $A(\gamma) \succ 0$ for all $\gamma \in \mathcal{C}$
  - γ^* is optimizer of dual problem, there exist (slow) algorithms $\gamma^{(i)} \to \gamma^*$

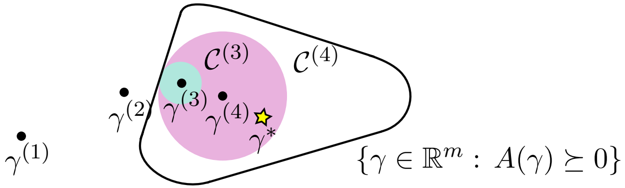


- How to construct $\mathcal{C}$? $\gamma^* \in \mathcal{C}$, $A(\gamma) \succ 0$ for all $\gamma \in \mathcal{C}$
  - γ^* is optimizer of dual problem, there exist (slow) algorithms $\gamma^{(i)} \to \gamma^*$

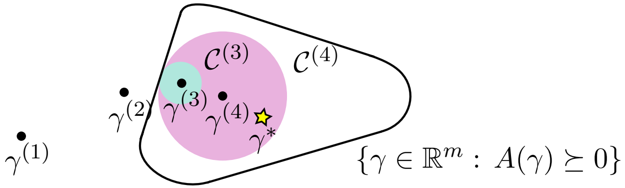


Algorithms: CertSDP

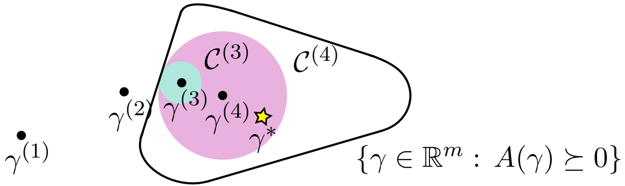
- How to construct $\mathcal{C}$? $\gamma^* \in \mathcal{C}$, $A(\gamma) \succ 0$ for all $\gamma \in \mathcal{C}$
  - γ^* is optimizer of dual problem, there exist (slow) algorithms $\gamma^{(i)} \to \gamma^*$



- How to construct $\mathcal{C}$? $\gamma^* \in \mathcal{C}$, $A(\gamma) \succ 0$ for all $\gamma \in \mathcal{C}$
  - γ^* is optimizer of dual problem, there exist (slow) algorithms $\gamma^{(i)} \to \gamma^*$
  - **Issue**: How do we know close enough?

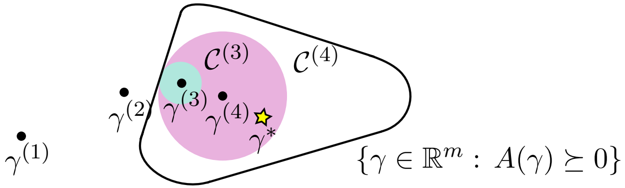


- How to construct $\mathcal{C}$? $\gamma^* \in \mathcal{C}$, $A(\gamma) \succ 0$ for all $\gamma \in \mathcal{C}$
  - γ^* is optimizer of dual problem, there exist (slow) algorithms $\gamma^{(i)} \to \gamma^*$
  - **Issue**: How do we know close enough?
    Use CautiousAGD!



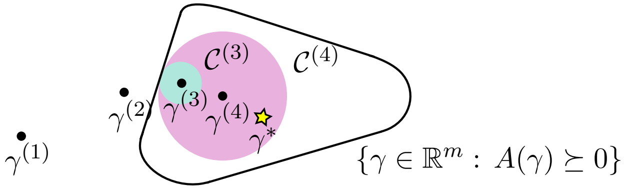
Algorithms: CertSDP

- How to construct $\mathcal{C}$? $\gamma^* \in \mathcal{C}$, $A(\gamma) \succ 0$ for all $\gamma \in \mathcal{C}$
  - γ^* is optimizer of dual problem, there exist (slow) algorithms $\gamma^{(i)} \to \gamma^*$
  - **Issue**: How do we know close enough?
    Use CautiousAGD!
    - If $\gamma^* \in \mathcal{C}^{(i)}$ then CautiousAGD converges to X^* rapidly

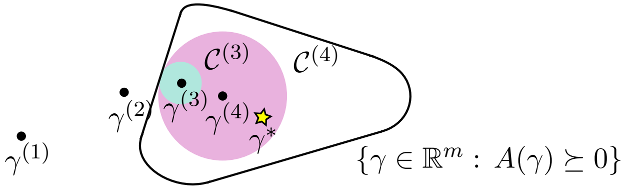


Algorithms: CertSDP

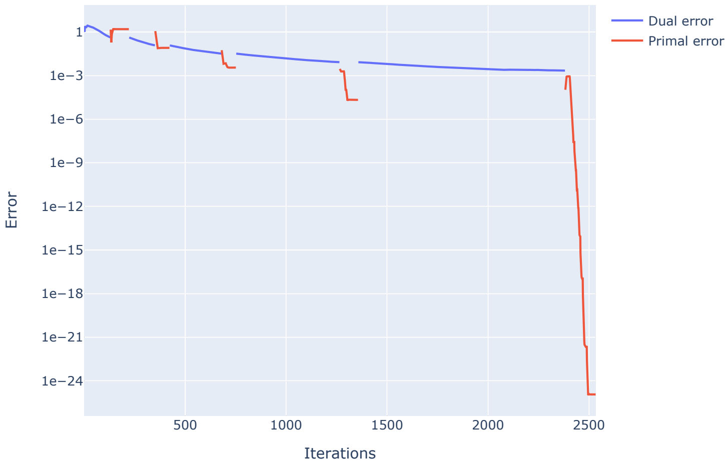
- How to construct $\mathcal{C}$? $\gamma^* \in \mathcal{C}$, $A(\gamma) \succ 0$ for all $\gamma \in \mathcal{C}$
  - γ^* is optimizer of dual problem, there exist (slow) algorithms $\gamma^{(i)} \to \gamma^*$
  - **Issue**: How do we know close enough?
    Use CautiousAGD!
    - If $\gamma^* \in \mathcal{C}^{(i)}$ then CautiousAGD converges to X^* rapidly
    - Monitor convergence!



- How to construct $\mathcal{C}$? $\gamma^* \in \mathcal{C}$, $A(\gamma) \succ 0$ for all $\gamma \in \mathcal{C}$
  - γ^* is optimizer of dual problem, there exist (slow) algorithms $\gamma^{(i)} \to \gamma^*$
  - **Issue**: How do we know close enough?
    Use CautiousAGD!
    - If $\gamma^* \in \mathcal{C}^{(i)}$ then CautiousAGD converges to X^* rapidly
    - Monitor convergence!
  - $\longrightarrow$ CertSDP



CertSDP convergence behavior



Theorem (CertSDP)

CertSDP produces iterates X_t such that

$$\left\langle M_0, \begin{pmatrix} X_t X_t^\intercal & X_t \\ X_t^\intercal & I_k \end{pmatrix} \right\rangle \leq \mathrm{Opt}_{\mathsf{SDP}} + \epsilon \qquad \left\| \left(\left\langle M_i, \begin{pmatrix} X_t X_t^\intercal & X_t \\ X_t^\intercal & I_k \end{pmatrix} \right\rangle + d_i \right)_i \right\|_2 \leq \epsilon$$

Related: Ding et al. [2021], Yurtsever et al. [2021], Friedlander and Macêdo [2016], Shinde et al. [2021]

CertSDP guarantees

Theorem (CertSDP)

CertSDP produces iterates X_t such that

$$\left\langle M_0, \begin{pmatrix} X_tX_t^\intercal & X_t \\ X_t^\intercal & I_k \end{pmatrix} \right\rangle \leq \mathrm{Opt}_{\mathsf{SDP}} + \epsilon \qquad \left\| \left(\left\langle M_i, \begin{pmatrix} X_tX_t^\intercal & X_t \\ X_t^\intercal & I_k \end{pmatrix} \right\rangle + d_i \right)_i \right\|_2 \leq \epsilon$$

- **Iteration count**: $O(1) + O\left(\log(\epsilon^{-1})\right)$

Related: Ding et al. [2021], Yurtsever et al. [2021], Friedlander and Macêdo [2016], Shinde et al. [2021]

CertSDP guarantees

Theorem (CertSDP)

CertSDP produces iterates X_t such that

$$\left\langle M_0, \begin{pmatrix} X_t X_t^\intercal & X_t \\ X_t^\intercal & I_k \end{pmatrix} \right\rangle \leq \mathrm{Opt}_{\mathsf{SDP}} + \epsilon \qquad \left\| \left(\left\langle M_i, \begin{pmatrix} X_t X_t^\intercal & X_t \\ X_t^\intercal & I_k \end{pmatrix} \right\rangle + d_i \right)_i \right\|_2 \leq \epsilon$$

- **Iteration count**: $O(1) + O\left(\log(\epsilon^{-1})\right)$
- **Iteration complexity**: $O(m\epsilon^{-1})$ matrix-vector products per iteration

Related: Ding et al. [2021], Yurtsever et al. [2021], Friedlander and Macêdo [2016], Shinde et al. [2021]

CertSDP guarantees

Theorem (CertSDP)

CertSDP produces iterates X_t such that

$$\left\langle M_0, \begin{pmatrix} X_t X_t^\intercal & X_t \\ X_t^\intercal & I_k \end{pmatrix} \right\rangle \leq \mathrm{Opt}_{\mathsf{SDP}} + \epsilon \qquad \left\| \left(\left\langle M_i, \begin{pmatrix} X_t X_t^\intercal & X_t \\ X_t^\intercal & I_k \end{pmatrix} \right\rangle + d_i \right)_i \right\|_2 \leq \epsilon$$

- **Iteration count**: $O(1) + O\left(\log(\epsilon^{-1})\right)$
- **Iteration complexity**: $O(m\epsilon^{-1})$ matrix-vector products per iteration
- **Storage**: $O(m + nk)$ storage

Related: Ding et al. [2021], Yurtsever et al. [2021], Friedlander and Macêdo [2016], Shinde et al. [2021]

1. What structure? . . . k-exactness
2. How to use k-exactness? . . . Strongly convex reformulation
3. Algorithms
4. Conclusion

Conclusion

- **Summary**: New FOM for k-exact SDPs

Conclusion

- **Summary**: New FOM for k-exact SDPs
  - Low iteration complexity, cheap iterations, and low storage requirement

Conclusion

- **Summary**: New FOM for k-exact SDPs
  - Low iteration complexity, cheap iterations, and low storage requirement
- **Future directions**:

Conclusion

- **Summary**: New FOM for k-exact SDPs
  - Low iteration complexity, cheap iterations, and low storage requirement
- **Future directions**:
  - Removing structural assumptions

Conclusion

- **Summary**: New FOM for k-exact SDPs
  - Low iteration complexity, cheap iterations, and low storage requirement
- **Future directions**:
  - Removing structural assumptions
    - Predicated on knowing a rank k submatrix of Y^* that is correlated with Y^*

Conclusion

- **Summary**: New FOM for k-exact SDPs
  - Low iteration complexity, cheap iterations, and low storage requirement
- **Future directions**:
  - Removing structural assumptions
    - Predicated on knowing a rank k submatrix of Y^* that is correlated with Y^*
    - “Learn” the subspace and the submatrix as part of the algorithm

Conclusion

- **Summary**: New FOM for k-exact SDPs
  - Low iteration complexity, cheap iterations, and low storage requirement
- **Future directions**:
  - Removing structural assumptions
    - Predicated on knowing a rank k submatrix of Y^* that is correlated with Y^*
    - "Learn" the subspace and the submatrix as part of the algorithm
    - Ongoing work

Conclusion

- **Summary**: New FOM for k-exact SDPs
  - Low iteration complexity, cheap iterations, and low storage requirement
- **Future directions**:
  - Removing structural assumptions
    - Predicated on knowing a rank k submatrix of Y^* that is correlated with Y^*
    - "Learn" the subspace and the submatrix as part of the algorithm
    - Ongoing work

Thank you! Questions?

`https://arxiv.org/abs/2206.00224`

Alizadeh, F., Haeberly, J. A., and Overton, M. L. (1997). Complementarity and nondegeneracy in semidefinite programming. *Math. Program.*, 77:111–128.

Beck, A. (2007). Quadratic matrix programming. *SIAM J. Optim.*, 17(4):1224–1238.

Beck, A., Drori, Y., and Teboulle, M. (2012). A new semidefinite programming relaxation scheme for a class of quadratic matrix problems. *Oper. Res. Lett.*, 40(4):298–302.

Ben-Tal, A. and Nemirovski, A. (2001). *Lectures on Modern Convex Optimization*, volume 2 of *MPS-SIAM Ser. Optim.*

Devolder, O., Glineur, F., and Nesterov, Y. (2013). First-order methods with inexact oracle: the strongly convex case. Technical Report 2013016.

Devolder, O., Glineur, F., and Nesterov, Y. (2014). First-order methods of smooth convex optimization with inexact oracle. *Math. Program.*, 146(1):37–75.

Ding, L., Yurtsever, A., Cevher, V., Tropp, J. A., and Udell, M. (2021). An optimal-storage approach to semidefinite programming using approximate complementarity. *SIAM J. Optim.*, 31(4):2695–2725.

Friedlander, M. P. and Macêdo, I. (2016). Low-rank spectral optimization via gauge duality. *SIAM Journal on Scientific Computing*, 38(3):A1616–A1638.

References II

Nesterov, Y. (2005). Excessive gap technique in nonsmooth convex minimization. *SIAM J. Optim.*, 16(1):235–249.

Nesterov, Y. (2018). *Lectures on convex optimization*. Number 137 in Springer Optim. and its Appl. 2 edition.

Shinde, N., Narayanan, V., and Saunderson, J. (2021). Memory-efficient structured convex optimization via extreme point sampling. *SIAM Journal on Mathematics of Data Science*, 3(3):787–814.

Wang, A. L. and Kılınç-Karzan, F. (2020). A geometric view of SDP exactness in QCQPs and its applications. *arXiv preprint*, 2011.07155.

Wang, A. L. and Kılınç-Karzan, F. (2022). On the tightness of SDP relaxations of QCQPs. *Math. Program.*, 193:33–73.

Yurtsever, A., Tropp, J. A., Fercoq, O., Udell, M., and Cevher, V. (2021). Scalable semidefinite programming. *SIAM Journal on Mathematics of Data Science*, 3(1):171–200.