An Optimal-Storage Approach to Semidefinite Programming using Approximate Complementarity

Lijun Ding

Joint work with Alp Yurtsever, Volkan Cevher, Joel A. Tropp and Madeleine Udell

March 9, 2019
1 Introduction
- Setup
- Related Work
- A Conceptual Approach

2 Robust Primal Recovery
- Problem of Conceptual approach
- Robust Recovery

3 Numerics
- Numerics Setup
- Numerical Results
SDP in standard form

- **Primal:**

  \[
  \begin{aligned}
  \text{minimize} & \quad \text{tr}(CX) \\
  \text{subject to} & \quad AX = b \quad \text{and} \quad X \succeq 0,
  \end{aligned}
  \tag{P}
  \]

  with problem data: a cost matrix \( C \in \mathbb{S}^n \), a righthand side \( b \in \mathbb{R}^m \),

  and a linear map \( A : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^m \).

- Assumption: the pair \((P)\) and \((D)\) admits unique solution \((X^\star, y^\star)\) and strong duality:

  \[ \text{tr}(CX^\star) = b^\ast y^\star. \]
SDP in standard form

- **Primal:**
  \[
  \begin{array}{ll}
  \text{minimize} & \text{tr}(CX) \\
  \text{subject to} & AX = b \quad \text{and} \quad X \succeq 0,
  \end{array}
  \]  
  \( (P) \)

  with problem data: a cost matrix \( C \in \mathbb{S}^n \), a righthand side \( b \in \mathbb{R}^m \), and a linear map \( A : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^m \).

- **Dual:**
  \[
  \begin{array}{ll}
  \text{maximize} & b^* y \\
  \text{subject to} & C - A^* y \succeq 0
  \end{array}
  \]  
  \( (D) \)

  where \( b^* \) is the transpose of \( b \), and the linear map \( A^* : \mathbb{R}^m \rightarrow \mathbb{R}^{n \times n} \) is the adjoint of the linear map \( A \).
SDP in standard form

- **Primal:**
  \[
  \begin{align*}
  \text{minimize} & \quad \sum_{i,j} x_{i,j} C_{i,j} \\
  \text{subject to} & \quad \sum_{i,j} A_{i,j} x_{i,j} = b \\
  & \quad x_{i,j} \geq 0, \quad i,j = 1, \ldots, n,
  \end{align*}
  \]
  with problem data: a cost matrix \( C \in \mathbb{S}^n \), a righthand side \( b \in \mathbb{R}^m \), and a linear map \( A : \mathbb{R}^{n \times n} \to \mathbb{R}^m \).

- **Dual:**
  \[
  \begin{align*}
  \text{maximize} & \quad \sum_{i,j} y_{i,j} b_{i,j} \\
  \text{subject to} & \quad \sum_{i,j} (C - A^* y)_{i,j} \geq 0
  \end{align*}
  \]
  where \( b^* \) is the transpose of \( b \), and the linear map \( A^* : \mathbb{R}^m \to \mathbb{R}^{n \times n} \) is the adjoint of the linear map \( A \).

**Assumption:** the pair (P) and (D) admits *unique* solution \((x_*, y_*)\) and *strong duality*:
\[
\text{tr}(CX_*) = b^* y_*.
\]
Motivation: Writing $X$ requires too much memory, e.g., $\mathcal{O}(10^{12})$ for a $10^6$ object in phase retrieval.
Motivation: Writing $X$ requires too much memory, e.g., $\mathcal{O}(10^{12})$ for a $10^6$ object in phase retrieval.

Operations allowed to interact with the problem data $A$ and $C$: 
$\forall u, v \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$:

$$u \mapsto Cu \quad \text{and} \quad (u, v) \mapsto A(\text{uv}^*) \quad \text{and} \quad (u, y) \mapsto (A^*y)u. \quad (1)$$
Storage Optimality

- Motivation: Writing $X$ requires too much memory, e.g., $O(10^{12})$ for a $10^6$ object in phase retrieval.

- Operations allowed to interact with the problem data $A$ and $C$:
  $\forall u, v \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$:
  \[ u \mapsto Cu \quad \text{and} \quad (u, v) \mapsto A(uv^*) \quad \text{and} \quad (u, y) \mapsto (A^*y)u. \quad (1) \]

- Any algorithm needs $\Theta(m + n)$ storage.
Storage Optimality

- Motivation: Writing $X$ requires too much memory, e.g., $O(10^{12})$ for a $10^6$ object in phase retrieval.
- Operations allowed to interact with the problem data $A$ and $C$: 
  \[ \forall u, v \in \mathbb{R}^n \text{ and } y \in \mathbb{R}^m: \]
  \[ u \mapsto Cu \quad \text{and} \quad (u, v) \mapsto A(uv^*) \quad \text{and} \quad (u, y) \mapsto (A^*y)u. \quad (1) \]
- Any algorithm needs $\Theta(m + n)$ storage.
- To output the rank $r^* = \text{rank}(X^*)$ solution $X^*$ needs $nr^*$ storage.
Storage Optimality

- Motivation: Writing $X$ requires too much memory, e.g., $O(10^{12})$ for a $10^6$ object in phase retrieval.

- Operations allowed to interact with the problem data $A$ and $C$:
  $\forall u, v \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$:
  
  $$u \mapsto Cu \quad \text{and} \quad (u, v) \mapsto A(uv^*) \quad \text{and} \quad (u, y) \mapsto (A^*y)u. \quad (1)$$

- Any algorithm needs $\Theta(m + n)$ storage.

- To output the rank $r_* = \text{rank}(X_*)$ solution $X_*$ needs $nr_*$ storage.

Storage Optimal

A method has *optimal storage* if the working storage is $O(m + nr_*)$ and access the data only through (1).
Why storage optimality interesting?

Concrete examples, maxcut relaxation [Goemans and Williamson 1995], and matrix completion [Srebro and Shraibman 2005], satisfies
Concrete examples, maxcut relaxation [Goemans and Williamson1995], and matrix completion [Srebro and Shraibman2005], satisfies

- Problem data $A$, $b$, $C$ only requires $O(n)$ storage with $m = O(n)$
- $r_\star$ is constant with respect to $n$
Why storage optimality interesting?

Concrete examples, maxcut relaxation [Goemans and Williamson1995], and matrix completion [Srebro and Shraibman2005], satisfies

- Problem data $A, b, C$ only requires $O(n)$ storage with $m = O(n)$
- $r_\star$ is constant with respect to $n$
- Storage optimality means $O(m + nr_\star) = O(n)$ storage!
Concrete examples, maxcut relaxation [Goemans and Williamson1995], and matrix completion [Srebro and Shraibman2005], satisfies

- Problem data $A, b, C$ only requires $O(n)$ storage with $m = O(n)$
- $r_\star$ is constant with respect to $n$
- Storage optimality means $O(m + nr_\star) = O(n)$ storage!
Concrete examples, maxcut relaxation [Goemans and Williamson1995], and matrix completion [Srebro and Shraibman2005], satisfies

- Problem data $A, b, C$ only requires $O(n)$ storage with $m = O(n)$
- $r_\star$ is constant with respect to $n$
- Storage optimality means $O(m + nr_\star) = O(n)$ storage!

**Solving ($\mathcal{P}$) without even writing $X$ down!**
Related Work: an incomplete list

- **Other First Order methods**: Spectral bundle method [Helmberg and Rendl 2000], Spectral low rank optimization [Friedlander and Macedo 2016], and Radial projection method [Renegar 2014].
- Lack convergence guarantees, not storage optimal, or do not apply to general (P).
Burer-Monteiro approach: Solving

\[
\begin{align*}
\text{minimize} & \quad \text{tr}(CFF^*) \\
\text{subject to} & \quad A(FF^*) = b, \quad F \in \mathbb{R}^{n \times r} \\
\end{align*}
\] (BM)
Related Work

**Burer-Monteiro approach:** Solving

\[
\begin{align*}
\text{minimize} & \quad \text{tr}(CFF^*) \\
\text{subject to} & \quad A(FF^*) = b, \quad F \in \mathbb{R}^{n \times r}
\end{align*}
\]  

\( (BM) \)

- Good side:

\[
\frac{r(r + 1)}{2} > m
\]

 guarantees second order stationary implies global optimality [Burer and Monteiro 2003], [Boumal et al. 2016].
Burer-Monteiro approach: Solving

\[
\begin{align*}
\text{minimize} & \quad \text{tr}(CFF^*) \\
\text{subject to} & \quad A(FF^*) = b, F \in \mathbb{R}^{n \times r}
\end{align*}
\]  

(BM)

- **Good side:**
  \[\frac{r(r + 1)}{2} > m\]
  guarantees second order stationary implies global optimality
  [Burer and Monteiro2003], [Boumal et al.2016].

- **Limitation:** [Waldspurger and Waters2018] shows that (BM) admits bad local minima for
  \[\frac{r(r + 1)}{2} + r \leq m\]
  for some \(A, b, C\).
This work

A new algorithm that *provably* solves the SDP (P) with *optimal storage*. 
Solving the dual

- Note the dual variable $y$ only occupies $m$ storage
Solving the dual

- Note the dual variable $y$ only occupies $m$ storage
- Penalized version:

$$
\text{maximize } b^* y + \alpha \min \{ \lambda_{\min}(C - A^* y), 0 \}. \quad (2)
$$
Solving the dual

Note the dual variable $y$ only occupies $m$ storage

Penalized version:

$$\text{maximize} \quad b^* y + \alpha \min \{\lambda_{\min}(C - A^* y), 0\}. \quad (2)$$

Equivalence: If $\alpha > \text{tr}(X_*)$, then the above is exact [Ding et al.2019, Lemma 6.1].
Solving the dual

- Note the dual variable $y$ only occupies $m$ storage

Penalized version:

$$\text{maximize} \quad b^* y + \alpha \min \{ \lambda_{\min}(C - A^* y), 0 \}. \quad (2)$$

- Equivalence: If $\alpha > \text{tr}(X_*)$, then the above is exact [Ding et al.2019, Lemma 6.1].

- Apply subgradient-type method. Computing eigenvectors via iterative methods.
Suppose we have the optimal dual $y_\star$ and $Z_\star = C - A^* y_\star$
Suppose we have the optimal dual $y_*$ and $Z_*=C-A^*y_*$

Complementary slackness:

$$X_*Z_* = 0 \iff \text{range}(X_*) \subset \text{null}Z_*$$
Suppose we have the optimal dual $y_\star$ and $Z_\star = C - A^* y_\star$

Complementary slackness:

$$X_\star Z_\star = 0 \implies \text{range}(X_\star) \subset \text{null} Z_\star$$

Assume strict complementarity:

$$\text{rank}(X_\star) + \text{rank}(Z_\star) = n \implies \text{range}(X_\star) = \text{null} Z_\star$$
Conceptual Primal Recovery

- Suppose we have the optimal dual $y_*$ and $Z_* = C - A^* y_*$
- Complementary slackness:

\[ X_* Z_* = 0 \implies \text{range}(X_*) \subseteq \text{null} Z_* \]

- Assume strict complementarity:

\[ \text{rank}(X_*) + \text{rank}(Z_*) = n \implies \text{range}(X_*) = \text{null} Z_* \]

- Compute an unitary representation $V \in \mathbb{R}^{n \times r_*}$ of $\text{null} Z_*$. 
Suppose we have the optimal dual $y_\star$ and $Z_\star = C - A^* y_\star$

Complementary slackness:

$$X_\star Z_\star = 0 \implies \text{range}(X_\star) \subset \text{null} Z_\star$$

Assume strict complementarity:

$$\text{rank}(X_\star) + \text{rank}(Z_\star) = n \implies \text{range}(X_\star) = \text{null} Z_\star$$

Compute an unitary representation $V \in \mathbb{R}^{n \times r_\star}$ of $\text{null} Z_\star$.

Solve the reduced SDP:

$$\begin{align*}
\text{minimize} & \quad \text{tr}(CV_\star S V_\star^*) \\
\text{subject to} & \quad A(V_\star S V_\star^*) = b \quad \text{and} \quad S \in S^{r_\star}_+
\end{align*}$$

(Reduced SDP)
A Conceptual Optimal-Storage Approach

Three steps:

1. Compute dual solution $y^*$

Optimal storage:

1. Subgradient method $O(m + n)$
2. Compute $V^*$ via the randomized range finder [Halko et al. 2011, Alg. 4.1] with storage cost $\Theta(nr^*)$
3. Solve Reduced SDP via the matrix-free method from [O’Donoghue et al. 2016] using $\Theta(m + n + r^2)$ storage.
A Conceptual Optimal-Storage Approach

Three steps:

1. Compute dual solution $y_*$
2. Compute basis $V_*$ for $\text{null}(C - A^*y_*)$
A Conceptual Optimal-Storage Approach

Three steps:

1. Compute dual solution $y_*$
2. Compute basis $V_*$ for $\text{null}(C - A^* y_*)$
3. Solve the Reduced SDP

\[
\begin{align*}
\text{minimize} & \quad \text{tr}(C V_* S V_*^*) \\
\text{subject to} & \quad A(V_* S V_*^*) = b \quad \text{and} \quad S \in S^r_+ \\
\end{align*}
\]  

(Reduced SDP)
A Conceptual Optimal-Storage Approach

Three steps:
1. Compute dual solution $y_*$
2. Compute basis $V_*$ for $\text{null}(C - A^*y_*)$
3. Solve the Reduced SDP

$$\begin{align*}
\text{minimize} & \quad \text{tr}(CV_*SV_*) \\
\text{subject to} & \quad A(V_*SV_*) = b \quad \text{and} \quad S \in S^r_+ \\
\end{align*}$$

(Reduced SDP)

Optimal storage:
4. Subgradient method $O(m + n)$
A Conceptual Optimal-Storage Approach

Three steps:

1. Compute dual solution $y_*$
2. Compute basis $V_*$ for $\text{null}(C - A^*y_*)$
3. Solve the Reduced SDP

$$\begin{align*}
\text{minimize} & \quad \text{tr}(CV_*SV_*) \\
\text{subject to} & \quad A(V_*SV_*) = b \quad \text{and} \quad S \in S^r_* \\
\end{align*}$$

(Reduced SDP)

Optimal storage:

1. Subgradient method $O(m + n)$
2. Compute $V_*$ via the randomized range finder [Halko et al. 2011, Alg. 4.1] with storage cost $\Theta(nr_*)$
A Conceptual Optimal-Storage Approach

Three steps:

1. Compute dual solution $y_\star$
2. Compute basis $V_\star$ for $\text{null}(C - A^* y_\star)$
3. Solve the Reduced SDP

\[
\begin{align*}
\text{minimize} \quad & \text{tr}(C V_\star S V_\star^*) \\
\text{subject to} \quad & A(V_\star S V_\star^*) = b \quad \text{and} \quad S \in S^r_+ \\
\end{align*}
\]

(Reduced SDP)

Optimal storage:

1. Subgradient method $\mathcal{O}(m + n)$
2. Compute $V_\star$ via the randomized range finder [Halko et al. 2011, Alg. 4.1] with storage cost $\Theta(nr_\star)$
3. solve Reduced SDP via the matrix-free method from [O’Donoghue et al. 2016] using $\Theta(m + n + r^2_\star)$ storage.
Outline

1. Introduction
   - Setup
   - Related Work
   - A Conceptual Approach

2. Robust Primal Recovery
   - Problem of Conceptual approach
   - Robust Recovery

3. Numerics
   - Numerics Setup
   - Numerical Results
Problems: Never get $y_\star$!

An easy fix?

Compute an approximate solution $y$

Compute $V$ formed by $r\star$ eigenvector of $C - A^*$

$y$

With smallest eigenvalues

Solve reduced SDP (Reduced SDP) with $V\star$ replaced by $V$:

$$\begin{align*}
\text{minimize} & \quad \text{tr}(CVSV^\star) \\
\text{subject to} & \quad A(VSV^\star) = b \\
& \quad S \in S_r^{\star} + (3)
\end{align*}$$

Status: Infeasible

Optimal value (cvx\_optval): $+\infty$
Problems: Never get $y_!$

Never get $y_!$

An easy fix?

- Compute an approximate solution $y$
- Compute $V$ formed by $r_*^*$ eigenvector of $C - A^* y$ with smallest eigenvalues
- Solve reduced SDP (Reduced SDP) with $V^*$ replaced by $V$:

$$\begin{align*}
\text{minimize} & \quad \text{tr}(CVSV^*) \\
\text{subject to} & \quad A(VSV^*) = b \quad \text{and} \quad S \in \mathbf{S}_+^{r_*}
\end{align*}$$

(3)
Problems: Never get $y_*$!

An easy fix?

- Compute an approximate solution $y$
- Compute $V$ formed by $r_\star$ eigenvector of $C - A^* y$ with smallest eigenvalues
- Solve reduced SDP (Reduced SDP) with $V_\star$ replaced by $V$:

$$\begin{align*}
\text{minimize} & \quad \text{tr}(C V S V^*) \\
\text{subject to} & \quad A(V S V^*) = b \quad \text{and} \quad S \in \mathbf{S}_+^{r_\star}
\end{align*}$$

Status: Infeasible
Optimal value (cvx_optval): $+\infty$
Choose an integer \( r \) (not necessarily \( r_\star \))

- Compute an approximate solution \( y \)
- Compute \( V \) formed by \( r \) eigenvector of \( C - A^*y \) with smallest eigenvalues
Choose an integer $r$ (not necessarily $= r_*$)

- Compute an approximate solution $y$
- Compute $V$ formed by $r$ eigenvector of $C - \mathcal{A}^* y$ with smallest eigenvalues
- Solve

\[
\begin{align*}
& \text{minimize} & & \frac{1}{2} \| \mathcal{A} (VSV^*) - b \|^2 \\
& \text{subject to} & & S \in \mathbf{S}_r^+, \\
\end{align*}
\]  

$\text{(MinFeasSDP)}$
Choose an integer $r$ (not necessarily $= r_*$)

- Compute an approximate solution $y$
- Compute $V$ formed by $r$ eigenvector of $C - A^*y$ with smallest eigenvalues
- Solve

$$\begin{align*}
\text{minimize} & \quad \frac{1}{2} \|A(VSV^*) - b\|^2 \\
\text{subject to} & \quad S \in S_r^+, \\
\end{align*}$$

(MinFeasSDP)

or choose a tolerance level $\delta$ and solve

$$\begin{align*}
\text{minimize} & \quad \text{tr}(CVSV^*) \\
\text{subject to} & \quad \|A(VSV^*) - b\| \leq \delta \quad \text{and} \quad S \in S_r^+, \\
\end{align*}$$

(MinObjSDP)
Comparison between Conceptual and Robust Approach

<table>
<thead>
<tr>
<th>Step</th>
<th><strong>Exact Primal Recovery</strong></th>
<th><strong>Robust Primal Recovery</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Compute dual solution $y_\star$</td>
<td>Compute approximate dual solution $y$</td>
</tr>
<tr>
<td>2</td>
<td>Compute basis $V_\star$ for $\text{null}(C - A^*y_\star)$</td>
<td>Compute $r$ eigenvectors of $C - A^*y$ with smallest eigenvalues; collect as columns of matrix $V$</td>
</tr>
<tr>
<td>3</td>
<td>Solve the Reduced SDP</td>
<td>Solve MinFeasSDP or MinObjSDP.</td>
</tr>
</tbody>
</table>
Define the condition number $\kappa = \frac{\sigma_{\text{max}}(A)}{\sigma_{\text{min}}(A|\mathbf{v}_x)}$ and assume strict complementarity, uniqueness and strong duality.
Theoretical Guarantees: MinFeasSDP and MinObjSDP

Define the condition number \( \kappa = \frac{\sigma_{\text{max}}(A)}{\sigma_{\text{min}}(A|V_\star)} \) and assume strict complementarity, uniqueness and strong duality.

**Table:** Comparison of the solution of MinFeasSDP and MinObjSDP given a feasible \( \epsilon \)-suboptimal dual vector \( y \), \( b^*y_\star - b^*y \leq \epsilon \).

<table>
<thead>
<tr>
<th>Require ( r = r_\star ) ?</th>
<th>MinFeasSDP</th>
<th>MinObjSDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Require ( r = r_\star ) ?</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Suboptimality: ( \text{tr}(CX) - \text{tr}(CX_\star) )</td>
<td>( O(\kappa \sqrt{\epsilon}) )</td>
<td>( O(\sqrt{\epsilon}) )</td>
</tr>
<tr>
<td>Infeasibility: ( |AX - b|_2 )</td>
<td>( O(\kappa \sqrt{\epsilon}) )</td>
<td>( O(\sqrt{\epsilon}) )</td>
</tr>
<tr>
<td>Distance to the solution: ( |X - X_\star|_F )</td>
<td>( O(\kappa \sqrt{\epsilon}) )</td>
<td>( O(\epsilon^{\frac{1}{4}}) )</td>
</tr>
</tbody>
</table>
Outline

1 Introduction
   - Setup
   - Related Work
   - A Conceptual Approach

2 Robust Primal Recovery
   - Problem of Conceptual approach
   - Robust Recovery

3 Numerics
   - Numerics Setup
   - Numerical Results
### Problem Instances:

<table>
<thead>
<tr>
<th>Max-Cut</th>
<th>Matrix Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimize $\text{tr}(-LX)$</td>
<td>minimize $\text{tr} \left( W_1 \right) + \text{tr} \left( W_2 \right)$</td>
</tr>
<tr>
<td>subject to (\text{diag}(X) = 1)</td>
<td>subject to $X_{ij} = \bar{X}_{ij}$, $(i, j) \in \Omega$</td>
</tr>
<tr>
<td>$X \succeq 0$</td>
<td>$\begin{bmatrix} W_1 &amp; X \ X^* &amp; W_2 \end{bmatrix} \succeq 0$</td>
</tr>
</tbody>
</table>
Problem Instances and Algorithm

Problem Instances:

<table>
<thead>
<tr>
<th>Max-Cut</th>
<th>Matrix Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimize $\text{tr}(-LX)$</td>
<td>minimize $\text{tr}(W_1) + \text{tr}(W_2)$</td>
</tr>
<tr>
<td>subject to $\text{diag}(X) = 1$</td>
<td>subject to $X_{ij} = \bar{X}_{ij}, (i,j) \in \Omega$</td>
</tr>
<tr>
<td>$X \succeq 0$</td>
<td>$\begin{bmatrix} W_1 &amp; X \ X^* &amp; W_2 \end{bmatrix} \succeq 0$</td>
</tr>
</tbody>
</table>

Algorithm:

For $k = 1, 2, \ldots$ Do
- Use iterative dual solver (AdaGrad [Duchi et al. 2011], AdaNGD [Levy 2017], and AccelGrad [Levy et al. 2018]) and get its $k$-th iterate $y_k$
- Use $y_k$ and Robust Primal Recovery to recover the primal solution
Problem Instances and Algorithm

Problem Instances:

<table>
<thead>
<tr>
<th>Max-Cut</th>
<th>Matrix Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimize ( \text{tr}(-LX) )</td>
<td>minimize ( \text{tr}(W_1) + \text{tr}(W_2) )</td>
</tr>
<tr>
<td>subject to ( \text{diag}(X) = 1 )</td>
<td>subject to ( X_{ij} = \bar{X}_{ij}, (i,j) \in \Omega )</td>
</tr>
<tr>
<td>( X \succeq 0 )</td>
<td>( \begin{bmatrix} W_1 &amp; X \ X^* &amp; W_2 \end{bmatrix} \succeq 0 )</td>
</tr>
</tbody>
</table>

Algorithm:

For \( k = 1, 2, \ldots \) Do

- Use iterative dual solver (AdaGrad [Duchi et al. 2011], AdaNGD [Levy 2017], and AccelGrad [Levy et al. 2018]) and get its \( k \)-th iterate \( y_k \)
- Use \( y_k \) and Robust Primal Recovery to recover the primal solution
Numerical Result

Figure: Max-Cut

Figure: Matrix Completion

Solving SDP via Complementarity

March 9, 2019


References V


Lemma (Quadratic Growth)

Assume strong duality, uniqueness of solutions, and strict complementarity. For any dual feasible \( y \) with dual slack matrix \( Z(y) := C - A^*y \) and dual suboptimality \( \epsilon = \epsilon_d(y) = d_\star - b^*y \), we have

\[
\| (Z(y), y) - (Z_\star, y_\star) \| \leq \frac{1}{\sigma_{\min}(\mathcal{D})} \left[ \frac{\epsilon}{\lambda_{\min > 0}(X_\star)} + \sqrt{\frac{2\epsilon}{\lambda_{\min > 0}(X_\star)}} \| Z(y) \|_{\text{op}} \right],
\]

where the linear operator \( \mathcal{D} : \mathbb{S}^n \times \mathbb{R}^m \to \mathbb{S}^n \times \mathbb{S}^n \) is defined by

\[
\mathcal{D}(Z, y) := (Z - (U_\star U_\star^*) Z (U_\star U_\star^*), Z + A^*y).
\]

The orthonormal matrix \( U_\star \) is the orthogonal complement of \( V_\star \).
Suppose (P) and (D) admit solutions and satisfy strong duality. Further suppose \( y \in \mathbb{R}^m \) is feasible and \( \epsilon \)-suboptimal for the dual SDP (D). Assume that the threshold \( T := \lambda_{n-r}(C - A^*y) > 0 \). For any solution \( X_\star \) of the primal SDP (P),

\[
\|X_\star - VV^*X_\star VV^*\|_F \leq \frac{\epsilon}{T} + \sqrt{2\frac{\epsilon}{T} \|X_\star\|_{op}},
\]

and

\[
\|X_\star - VV^*X_\star VV^*\|_* \leq \frac{\epsilon}{T} + 2\sqrt{r \frac{\epsilon}{T} \|X_\star\|_{op}}.
\]
Lemma

Assume the same as Lemma 1. Then \( \text{null}(A_{V^*}) = \{0\} \).