Overview

CSDP is a solver for semidefinite programming (SDP) problems. It was originally written in 1997 and its C implementation continues to be periodically supported by the original author, Brian Borchers. Using the ASCII SDPA format[1], SDPs can be easily passed to the solver. Alternatively, problems can be directly encoded via its C API.

The purpose of this document is to give examples of how to reformulate SDPs into the form that CSDP accepts (either SDPA or its C interface). This often requires algebraic manipulation to write the problem in the primal SDP form as specified in the CSDP User Guide[2]. Note that CSDP can work with matrices that have both symmetric blocks and diagonal blocks. The use of diagonal blocks are particularly useful for encoding and solving linear programs (LP).

Semidefinite programming is a subfield of convex optimization. These problems are defined as optimizing a linear objective function with respect to linear constraints over the space of real symmetric $n \times n$ matrices, $\mathbb{S}^n$. CSDP solves semidefinite programming primal–dual problems of the form

$$\begin{aligned} \mathcal{P}: \quad & \max_{X \in \mathbb{S}^n} && \operatorname{tr}(CX) \\ & \text{s.t.} && \operatorname{tr}(A_i X) = b_i,\ \forall i \in [m] \\ & && X \succeq 0 \end{aligned}$$

$$\begin{aligned} \mathcal{D}: \quad & \min_{\substack{y \in \mathbb{R}^m \\ Z \in \mathbb{S}^n}} && b^\top y \\ & \text{s.t.} && \sum_{i=1}^m y_i A_i - C = Z \\ & && Z \succeq 0, \end{aligned}$$

where $C, A_1, \dots, A_m \in \mathbb{S}^n$, $y, b \in \mathbb{R}^m$. Note that different packages and presentations of primal–dual SDP problems may differ slightly; for example, the min and the max may be swapped, or the constraint in the dual may not have the explicit decision variable Z and instead write $\sum_{i=1}^m y_i A_i - C \succeq 0$.

To put SDPs in context with other convex optimizations, note the following hierarchy[3]

$$\text{LP} \subseteq \text{QP} \subseteq \text{QCQP} \subseteq \text{SOCP} \subseteq \text{SDP}.$$

Linear Matrix Inequalities

A linear matrix inequality (LMI) in variable $y \in \mathbb{R}^m$ has the form[4]

$$F(y) = F_0 + y_1 F_1 + \cdots + y_m F_m \succeq 0, \tag{1}$$

where $F_0, \dots, F_m \in \mathbb{S}^n$, and specifies a convex constraint on y. Note that for $\sum_i y_i A_i - C$ to equal $F(y)$, $F_0 := -C$. This small detail will likely cause sign errors in various places.

Many inequalities can be represented as LMIs, for example, for $d_i, z \in \mathbb{R}^N$ and $f \in \mathbb{R}^M$

$$d_i^\top z \le f_i,\ \forall i \in [M],$$

can be expressed as the LMI

$$\begin{bmatrix} f_1 - d_1^\top z & 0 & \cdots & 0 \\ 0 & f_2 - d_2^\top z & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & f_M - d_M^\top z \end{bmatrix} \succeq 0.$$

Similarly, if we wish to express the equality constraints

$$d_i^\top z = f_i,\ \forall i \in [M],$$

with variable $z \in \mathbb{R}^N$ as an LMI, we could split the constraints into the pairs of inequalities

$$d_i^\top z \le f_i, \quad d_i^\top z \ge f_i, \quad \forall i \in [M],$$

and use the same LMI form from above

$$\operatorname{diag}(\begin{bmatrix} f_1 - d_1^\top z & \cdots & f_M - d_M^\top z & d_1^\top z - f_1 & \cdots & d_M^\top z - f_M \end{bmatrix}) \succeq 0.$$

[1] `http://plato.asu.edu/ftp/sdpa_format.txt`
[2] `https://github.com/coin-or/Csdp/files/2485526/csdpuser.pdf`
[3] Princeton `ORF 523 Notes`
[4] `Stanford EE363 Notes`

Example

Let $z \in \mathbb{R}^2$ with data

$$D = \begin{bmatrix} 0 & 4 \\ 1 & -1 \\ 3 & 2 \end{bmatrix} \in \mathbb{R}^{3\times 2}$$
$$f = \begin{bmatrix} 8 & 7 & 9 \end{bmatrix}^\top \in \mathbb{R}^3.$$

We can extract the rows of D as $D^\top = \begin{bmatrix} d_1^\top & d_2^\top & d_3^\top \end{bmatrix} \in \mathbb{R}^{2\times 3}$. Writing out the constraints

$$\begin{aligned} 4z_2 &\le 8 \\ z_1 - z_2 &\le 7 \\ 3z_1 + 2z_2 &\le 9 \end{aligned} \quad \iff \quad \begin{aligned} 8 - 4z_2 &\ge 0 \\ 7 - z_1 + z_2 &\ge 0 \\ 9 - 3z_1 - 2z_2 &\ge 0 \end{aligned},$$

it is clear to see that an LMI can be written as above.

Reformulating an LP as an SDP

Any LP is a special instance of an SDP[5], in particular its dual (as defined in this document). Given the following LP

$$\begin{aligned} \min_{y\in\mathbb{R}^m} \quad & b^\top y \\ \text{s.t.} \quad & Ay + c \ge 0, \end{aligned} \tag{2}$$

where $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{n\times m}$, and $c \in \mathbb{R}^n$, it can be written as the dual SDP

$$\begin{aligned} \min_{y\in\mathbb{R}^m} \quad & b^\top y \\ \text{s.t.} \quad & F(y) \succeq 0, \end{aligned} \tag{3}$$

where $F(y) = F_0 + y_1 F_1 + \cdots + y_m F_m$ is an LMI. Given that $A = \begin{bmatrix} a_1 & \dots & a_m \end{bmatrix} \in \mathbb{R}^{n\times m}$, the definitions of F_i are as follows

$$\begin{aligned} F_0 &= \operatorname{diag}(c) \in \mathbb{S}^n \\ F_i &= \operatorname{diag}(a_i) \in \mathbb{S}^n, \ \forall i \in [m]. \end{aligned}$$

Note that

$$\begin{aligned} F(y) &= \operatorname{diag}(\{c_j + y_1 a_{1j} + \cdots + y_m a_{mj}\}_{j=1}^n) && (4) \\ &= \operatorname{diag}(c^\top + y^\top A^\top) = \operatorname{diag}(Ay + c). && (5) \end{aligned}$$

Example

Consider the following LP[6], with $m = 3$ and $n = 10$

$$\begin{aligned} \max_{p,q,r\in\mathbb{R}} \quad & z = 45p + 60q + 50r \\ \text{s.t.} \quad & 20p + 10q + 10r \le 2400 \\ & 12p + 28q + 16r \le 2400 \\ & 15p + 6q + 16r \le 2400 \\ & 20p + 15q \le 2400 \\ & 0 \le p \le 100 \\ & 0 \le q \le 40 \\ & 0 \le r \le 60 \end{aligned} \tag{6}$$

where the optimal solution is $p^* = 81.82$, $q^* = 16.36$, and $r^* = 60$. The optimal objective value is $z^* = 7664$.

[5]Slide 13 `CMU 10-725`
[6]Slide 16 of `MIT OCW 15.053 Lecure Notes`.

It turns out that an LP is really a special case of an SDP where all the matrices are diagonal (positive semidefiniteness for a diagonal matrix means nonnegativity of its diagonal elements). Our goal is to formulate this as an SDP so that we can use the CSDP solver. To do this, we observe that the LP (particularly, the objective) is most like the dual part of this primal–dual SDP pair (i.e., the primary decision variable of the dual is $y \in \mathbb{R}^n$, which is like our LP).

First, we massage the constraints of (6) into a form that looks more like the dual SDP

$$
\begin{aligned}
\min_{p,q,r\in\mathbb{R}} \quad & z = 45p + 60q + 50r \\
\text{s.t.} \quad & 20p + 10q + 10r + 2400 \geq 0 \\
& 12p + 28q + 16r + 2400 \geq 0 \\
& 15p + \ \ 6q + 16r + 2400 \geq 0 \\
& 20p + 15q \qquad\quad + 2400 \geq 0 \\
& \qquad\qquad\qquad p + 100 \geq 0 \, \cdot \\
& \qquad\qquad\qquad q + 40 \geq 0 \\
& \qquad\qquad\qquad r + 60 \geq 0 \\
& \qquad\qquad\qquad -p + 0 \geq 0 \\
& \qquad\qquad\qquad -q + 0 \geq 0 \\
& \qquad\qquad\qquad -r + 0 \geq 0
\end{aligned}
\tag{7}
$$

Note that we did three things: (1) rearranged the inequalities so that 0 is on the RHS; (2) flipped the sign of the objective so that the problem is a min. Simultaneously, we redefined p, q, and r to be their negative. This way, the resulting objective has plus signs and each term of the constraints (except the last three constraints) are negative; (3) flipped the sign of the inequalities by multiplying both sides by -1.

By pattern matching, we make the following assignments

$$
\begin{aligned}
y &= \begin{bmatrix} p & q & r \end{bmatrix}^\top \in \mathbb{R}^m; \qquad b = \begin{bmatrix} 40 & 60 & 50 \end{bmatrix}^\top \in \mathbb{R}^m \\
C &= -\operatorname{diag}(\begin{bmatrix} 2400 & 2400 & 2400 & 2400 & 100 & 40 & 60 & 0 & 0 & 0 \end{bmatrix}) \in \mathbb{S}^n \\
A_1 &= \operatorname{diag}(\begin{bmatrix} 20 & 12 & 15 & 20 & 1 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}) \in \mathbb{S}^n \\
A_2 &= \operatorname{diag}(\begin{bmatrix} 10 & 28 & 6 & 15 & 0 & 1 & 0 & 0 & -1 & 0 \end{bmatrix}) \in \mathbb{S}^n \\
A_2 &= \operatorname{diag}(\begin{bmatrix} 10 & 16 & 16 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}) \in \mathbb{S}^n.
\end{aligned}
$$

Thus, the following primal problem that encodes LP (6) can be input into CSDP[7]

$$
\begin{aligned}
\max_{X\in\mathbb{S}^n} \quad & \operatorname{tr}(CX) \\
\text{s.t.} \quad & \operatorname{tr}(A_1X) = b_1 = 45 \\
& \operatorname{tr}(A_2X) = b_2 = 60 \\
& \operatorname{tr}(A_2X) = b_3 = 50 \\
& X \succeq 0.
\end{aligned}
\tag{8}
$$

Note that CSDP provides the optimal values of X, y, and Z. The values we are interested in are in y, because we reformulated LP (6) as the dual.

Eigenvalue Optimization

Consider the task of minimizing the maximum eigenvalue of a symmetric matrix. This is useful in many cases, including optimizing for the characteristics of dynamical systems. Using the LMI $A(y)$, the problem can be stated as

$$
\min_{y\in\mathbb{R}^m} \quad \lambda_{\max}(A(y)) \; .
$$

Writing this as an SDP involves the epigraph formulation, where we introduce a new decision variable, $t \in \mathbb{R}$

$$
\begin{aligned}
\min_{y,\,t} \quad & t \\
\text{s.t.} \quad & \lambda_{\max}(A(y)) \leq t
\end{aligned}
\qquad \iff \qquad
\begin{aligned}
\min_{y,\,t} \quad & t \\
\text{s.t.} \quad & A(y) \preceq t
\end{aligned}
$$

By minimizing t, the max eigenvalue of $A(y)$ is constrained to decrease. Observe that we can use the language of semidefinite matrices to express the constraint as an SDP.

[7] `lp.cpp`

Example[8]

We can use an LMI to formulate the following max-eig minimization as an SDP

$$\min_{y\in\mathbb{R}^2} \quad \lambda_{\max}\left(\begin{bmatrix}1+y_1 & y_2-3\\ y_2-3 & -y+1+2y_2\end{bmatrix}\right) \quad \iff \quad \begin{aligned}\min_{y,\,t} \quad & t\\ \text{s.t.} \quad & \begin{bmatrix}1 & -3\\ -3 & 0\end{bmatrix} + y_1\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} + y_2\begin{bmatrix}0 & 1\\ 1 & 2\end{bmatrix} \preceq tI_2\end{aligned}.$$

Note that we can go all the way down to the standard dual form

$$\begin{aligned}\min_{y\in\mathbb{R}^m} \quad & b^\top y\\ \text{s.t.} \quad & \sum_{i=1}^m y_i A_i - C = Z \ ,\\ & Z \succeq 0\end{aligned}$$

by augmenting the decision variable vector $y = \begin{bmatrix}y_1 & y_2 & t\end{bmatrix}^\top \in \mathbb{R}^3$, flipping the p.s.d constraint direction, and making the following definitions

$$b = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^\top$$

$$A_1 = \begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix} \in \mathbb{S}^2 \qquad A_2 = \begin{bmatrix}0 & -1\\ -1 & -2\end{bmatrix} \in \mathbb{S}^2 \qquad A_3 = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \in \mathbb{S}^2 \qquad C = \begin{bmatrix}1 & -3\\ -3 & 0\end{bmatrix} \in \mathbb{S}^2.$$

For the sake of completeness, we give the SPDA sparse ASCII format of this problem, which can be read into CSDP via the command line[9].

```
"max-eig minimization problem
3 =m
1 =num blocks
2 =size of individual blocks
0 0 1 =bvec
0 1 1 1   1
0 1 1 2  -3
1 1 1 1  -1
1 1 2 2   1
2 1 1 2  -1
2 1 2 2  -2
3 1 1 1   1
3 1 2 2   1
```

Spectral Norm Minimization[10]

Given the LMI $A(y)$, we would like to minimize its maximum singular value, $\|A(y)\|_2 = \sigma_{\max}(A(y))$. This can be done via the following convex optimization

$$\min_{y\in\mathbb{R}^m} \quad \|A(y)\|_2. \tag{9}$$

As in the **Eigenvalue Optimization** section, we first rewrite the problem in epigraph form with the new decision variable, $s \in \mathbb{R}$

$$\begin{aligned}\min_{y,\,s} \quad & s\\ \text{s.t.} \quad & \|A(y)\|_2 \le s\end{aligned} \tag{10}$$

[8]Slide 12 `Purdue ECE695 Lecture`

[9]As opposed to above, which can be entered using the CSDP C API.

[10]`https://math.stackexchange.com/a/2137408/481663`

Focusing just on the constraint, we have the following equivalences

$$\begin{aligned}\{\|A(y)\|_2 \le s\} &= \{\sigma_{\max}(A(y)) \le s\} = \left\{\lambda_{\max}\left(A(y)^\top A(y)\right) \le s^2\right\} \\ &= \left\{s^2 - \lambda_{\max}\left(A(y)^\top A(y)\right) \ge 0\right\} \\ &= \left\{s^2 + \lambda_{\min}\left(-A(y)^\top A(y)\right) \ge 0\right\} \\ &= \left\{\lambda_{\min}\left(s^2 I_m - A(y)^\top A(y)\right) \ge 0\right\} \\ &= \left\{s^2 I_m - A(y)^\top A(y) \succeq 0\right\}.\end{aligned}$$

This last constraint definitely looks like what we want. However, while it is a convex positive semidefinite constraint, it is currently expressed as quadratic in the decision variables and needs to be reformulated to be an SDP. Dividing both sides by $s > 0$, we have

$$sI_m - A(y)^\top (sI_m)^{-1} A(y) \succeq 0. \tag{11}$$

It is only fitting[11] that a Schur complement would show up here. Recall[12] that given a matrix

$$X = \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \in \mathbb{R}^{k\times k},$$

if $A \in \mathbb{R}^{l\times l} \succ 0$, then $X \succeq 0 \iff S = C - B\top A^{-1} B \succeq 0$. Therefore, we can write constraint (11) as the LMI

$$\begin{bmatrix} sI_m & A(y) \\ A(y)^\top & sI_m \end{bmatrix} \succeq 0 \tag{12}$$

Note that in this analysis, we have not leveraged the symmetry of $A(y)$ (i.e., constraint (12) works if $A(y)$ is non-symmetric). If $A(y)$ is symmetric (as it is in this case since it is an LMI), we can formulate the constraint in a different way. First, observe[13] that for a symmetric matrix Z,

$$\sigma_{\max}(Z) = \max\{-\lambda_{\min}(Z), \lambda_{\max}(Z)\} \tag{13}$$

Therefore, if $A(y)$ is symmetric, we can immediately write problem (10) as

$$\begin{aligned} \min_{y,\, s} \quad & s \\ \text{s.t.} \quad & -sI_m \preceq A(y) \preceq sI_m \end{aligned} \tag{14}$$

which can be rewritten as

$$\begin{aligned} \min_{y,\, s} \quad & s \\ \text{s.t.} \quad & \begin{bmatrix} sI_m - A(y) & 0_m \\ 0_m & sI_m + A(y) \end{bmatrix} \succeq 0 \end{aligned}. \tag{15}$$

Note that (15) should be preferred over (12) because a pair of LMIs is more performant than one twice the size.

[11] It's a 'Schur' thing.
[12] Page 2, `Stanford EE363 Notes`
[13] Comments, `https://math.stackexchange.com/a/3075970/481663`