Semidefinite Programming,

Computational Intelligence, Lecture 10

by Sergei Savin

Spring 2021

Content

General form of a semidefinite program is:

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} \quad \mathbf{c}^\top \mathbf{x}, \\ &\text{subject to} \quad \begin{cases} \mathbf{G} + \sum \mathbf{F}_i x_i \preceq 0, \\ \mathbf{A}\mathbf{x} = \mathbf{b}. \end{cases} \end{aligned} \tag{1}$$

where $\mathbf{F}_i \succeq 0$ and $\mathbf{G} \succeq 0$ (meaning they are positive semidefinite).

Constraint $\mathbf{G} + \sum \mathbf{F}_i x_i \preceq 0$ is called *linear matrix inequality* or *LMI*.

SDP can have several LMIs. Assume you have:

$$\begin{cases} \mathbf{G} + \sum \mathbf{F}_i x_i \preceq 0 \\ \mathbf{D} + \sum \mathbf{H}_i x_i \preceq 0 \end{cases} \tag{2}$$

This is equivalent to:

$$\begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} + \sum \begin{bmatrix} \mathbf{F}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_i \end{bmatrix} x_i \preceq 0 \tag{3}$$

Sometimes it is easier to directly think of semidefinite matrices as of decision variables. This leads to programs with such formulation:

$$\begin{aligned} &\underset{\mathbf{X}}{\text{minimize}} \quad f(\mathbf{X}), \\ &\text{subject to} \quad \begin{cases} \mathbf{X} \preceq 0, \\ \mathbf{g}(\mathbf{X}) = \mathbf{0}. \end{cases} \end{aligned} \tag{4}$$

where cost and constraints should adhere to SDP limitations.

In control theory, Lyapunov equation is a condition of whether or not a continuous LTI system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$ is stabilizable:

$$
\begin{cases}
\mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{Q} = 0 \\
\mathbf{P} \succeq 0 \\
\mathbf{Q} \succeq 0
\end{cases}
\tag{5}
$$

where decision variable is $\mathbf{P}$. This can be represented as an SDP:

$$
\begin{aligned}
& \underset{\mathbf{P}}{\text{minimize}} && 0, \\
& \text{subject to} && \begin{cases}
\mathbf{P} \succeq 0, \\
\mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{Q} = 0.
\end{cases}
\end{aligned}
\tag{6}
$$

Ex. 1: Continuous Lyapunov eq. as SDP/LMI

Code

```
n = 7; A = randn(n, n) - 3*rand*eye(n);
Q = eye(n);

cvx_begin sdp
    variable P(n, n) symmetric
    minimize 0
    subject to
        P >= 0;
        A'*P + P*A + Q <= 0;
cvx_end

if strcmp(cvx_status, 'Solved')
    [eig(A), eig(A*P + P*A' + Q), eig(P)]
else
    eig(A)
end
```

In control theory, Discrete Lyapunov equation is a condition of whether or not a discrete LTI system $\mathbf{x}_{i+1} = \mathbf{A}\mathbf{x}_i$ is stabilizable:

$$\begin{cases} \mathbf{A}^\top \mathbf{P}\mathbf{A} - \mathbf{P} + \mathbf{Q} = 0 \\ \mathbf{P} \succeq 0 \\ \mathbf{Q} \succeq 0 \end{cases} \tag{7}$$

where decision variable is $\mathbf{P}$. This can be represented as an SDP:

$$\begin{aligned} \underset{\mathbf{P}}{\text{minimize}} \quad & 0, \\ \text{subject to} \quad & \begin{cases} \mathbf{P} \succeq 0, \\ \mathbf{A}^\top \mathbf{P}\mathbf{A} - \mathbf{P} + \mathbf{Q} = 0. \end{cases} \end{aligned} \tag{8}$$

Code

```
n = 7; A = 0.35*randn(n, n);
Q = eye(n);

cvx_begin sdp
    variable P(n, n) symmetric
    minimize 0
    subject to
        P >= 0;
        A'*P*A - P + Q <= 0;
cvx_end

if strcmp(cvx_status, 'Solved')
    [abs(eig(A)), eig(A'*P*A - P), eig(P)]
else
    abs(eig(A))
end
```

For an LTI system of the form $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ there is an LMI condition to determine if it can be stabilized:

$$\begin{cases} \mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^\top + \mathbf{B}\mathbf{L} + \mathbf{L}\mathbf{B}^\top + \mathbf{Q} = 0 \\ \mathbf{P} \succeq 0 \\ \mathbf{Q} \succeq 0 \end{cases} \tag{9}$$

where decision variables are $\mathbf{P}$ and $\mathbf{L}$.

This gives as a direct way to calculate linear feedback controller $\mathbf{u} = \mathbf{K}\mathbf{x}$ (note the sign!) gains:

$$\mathbf{K} = \mathbf{L}\mathbf{P}^{-1} \tag{10}$$

Ex. 3: FTS for continuous LTI, Code

```
n = 5; m = 2;
A = randn(n, n);
B = randn(n, m);
Q = eye(n)*0.1;
cvx_begin sdp
    variable P(n, n) symmetric
    variable Z(m, n)

    minimize 0
    subject to
      P >= 0;
        A*P + P*A' + B*Z + Z'*B' + Q <= 0;
cvx_end
P = full(P);
Z = full(Z);
K_LMI = Z*pinv(P);

disp('K_LMI eig:')
eig(A + B*K_LMI)
```

Let us first remember how we describe a unit sphere:

$$\mathcal{S} = \{\mathbf{x} : \ ||\mathbf{x}|| \leq 1\} \tag{11}$$

An ellipsoid can be seen as a linear transformation of a unit sphere:

$$\mathcal{E} = \{\mathbf{A}\mathbf{x} + \mathbf{b} : \ ||\mathbf{x}|| \leq 1\} \tag{12}$$

Let us introduce a change of variables $\mathbf{z} = \mathbf{A}\mathbf{x} + \mathbf{b}$. Assuming $\mathbf{A}$ is invertible, we get:

$$\mathbf{x} = \mathbf{A}^{-1}(\mathbf{z} - \mathbf{b}) \tag{13}$$

So, we can describe the exact same ellipsoid using an alternative formula:

$$\mathcal{E} = \{\mathbf{z} : \ ||\mathbf{B}\mathbf{z} + \mathbf{c}|| \leq 1\} \tag{14}$$

where $\mathbf{B} = \mathbf{A}^{-1}$ and $\mathbf{c} = -\mathbf{A}^{-1}\mathbf{b}$.

For an ellipsoid of the form

$$\mathcal{E} = \{\mathbf{A}\mathbf{x} + \mathbf{b} : \; ||\mathbf{x}|| \leq 1\} \tag{15}$$

the "bigger" the $\mathbf{A}$, the bigger the ellipsoid. This concept can be made concrete by talking about the determinant of $\mathbf{A}$.

Thus, maximizing the volume of this ellipsoid is the same as maximizing $\det(\mathbf{A})$. Or, it is the same as *minimizing* the $\det(\mathbf{A}^{-1})$, since $\det(\mathbf{A}^{-1}) = 1/\det(\mathbf{A})$.

For an ellipsoid of the form

$$\mathcal{E} = \{\mathbf{z} : \ ||\mathbf{B}\mathbf{z} + \mathbf{c}|| \leq 1\} \tag{16}$$

the "bigger" the $\mathbf{B}$, the *smaller* the ellipsoid. We can make it obvious by thinking that increasing $\mathbf{B}$ leaves less room for valid $\mathbf{z}$, and it is the volume of valid $\mathbf{z}$ that makes the volume of the ellipsoid in this case.

This concept can be made concrete by talking about the determinant of $\mathbf{B}$. Thus, maximizing the volume of this ellipsoid is the same as *minimizing* $\det(\mathbf{B})$. Or, it is the same as *maximizing* the $\det(\mathbf{B}^{-1})$.

Continue with slides from Convex Optimization — Boyd & Vandenberghe. Follow the link:

8. Geometric problems

Implement both examples from page 2 of the LMI CVX documents.

Lecture slides are available via Moodle.

You can help improve these slides at:
github.com/SergeiSa/Computational-Intelligence-Slides-Spring-2021

Check Moodle for additional links, videos, textbook suggestions.