Lecture 16: Semidefinite Programming and Duality Theorems

Rafael Oliveira

University of Waterloo
Cheriton School of Computer Science

rafael.oliveira.teaching@gmail.com

November 9, 2020

Overview

- Part I
  - Why Semidefinite Programming?
  - Convex Algebraic Geometry
- Part II
  - Duality Theory
  - Application: Control Theory
- Conclusion
- Acknowledgements

Mathematical Programming

Mathematical Programming deals with problems of the form

Mathematical Programming

Mathematical Programming deals with problems of the form

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_1(x) \geq 0 \\ & \quad \vdots \\ & g_m(x) \geq 0 \\ & x \in \mathbb{R}^n \end{aligned}$$

Mathematical Programming

Mathematical Programming deals with problems of the form

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_1(x) \geq 0 \\ & \quad \vdots \\ & g_m(x) \geq 0 \\ & x \in \mathbb{R}^n \end{aligned}$$

- Very general family of problems.

Mathematical Programming

Mathematical Programming deals with problems of the form

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_1(x) \geq 0 \\ & \vdots \\ & g_m(x) \geq 0 \\ & x \in \mathbb{R}^n \end{aligned}$$

- Very general family of problems.
- Special case when all $f, g_1, \ldots, g_m$ are *linear*. *Linear Programming*

Mathematical Programming

Mathematical Programming deals with problems of the form

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_1(x) \geq 0 \\ & \vdots \\ & g_m(x) \geq 0 \\ & x \in \mathbb{R}^n \end{aligned}$$

- Very general family of problems.
- Special case when all $f, g_1, \ldots, g_m$ are *linear*. *Linear Programming*
- More general case: *Semidefinite Programming*
    1. $A_1, \ldots, A_n, B \in \mathcal{S}^m$ are $m \times m$ symmetric matrices

Mathematical Programming

Mathematical Programming deals with problems of the form

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_1(x) \geq 0 \\ & \vdots \\ & g_m(x) \geq 0 \\ & x \in \mathbb{R}^n \end{aligned}$$

- Very general family of problems.
- Special case when all $f, g_1, \ldots, g_m$ are *linear*. *Linear Programming*
- More general case: *Semidefinite Programming*
  1. $A_1, \ldots, A_n, B \in \mathcal{S}^m$ are $m \times m$ symmetric matrices
  2. Constraints:
     $$x_1 \cdot A_1 + \cdots + x_n \cdot A_n \succeq B$$
  3. Minimize linear function $c^T x$

What is a Semidefinite Program?

- $\mathcal{S}^m := \mathcal{S}^m(\mathbb{R})$ space of all $m \times m$ symmetric matrices (real entries)

What is a Semidefinite Program?

- $\mathcal{S}^m := \mathcal{S}^m(\mathbb{R})$ space of all $m \times m$ symmetric matrices (real entries)
- $A \in \mathcal{S}^m$ is *positive semidefinite* (PSD), denoted $A \succeq 0$ if

$A_{ij} = A_{ji} \quad \forall \quad i,j \in [m]$

What is a Semidefinite Program?

- $\mathcal{S}^m := \mathcal{S}^m(\mathbb{R})$ space of all $m \times m$ symmetric matrices (real entries)
- $A \in \mathcal{S}^m$ is *positive semidefinite* (PSD), denoted $A \succeq 0$ if
  1. all eigenvalues of A are *non-negative*
  2. $A = Y^T Y$ for some $Y \in \mathbb{R}^{d \times m}$ where $d \leq m$
  3. $z^T A z \geq 0$ for any $z \in \mathbb{R}^m$
  4. and more...

$$x_1 A_1 + \cdots + x_n A_n \succeq B$$

$A(x_1, \ldots, x_n) = x_1 A_1 + \cdots + x_n A_n - B$ is PSD $\iff$

$$\underbrace{\lambda_1(A(\bar{x}))}_{g_1(\bar{x})}, \ldots, \underbrace{\lambda_m(A(\bar{x}))}_{g_m(\bar{x})} \geq 0$$

What is a Semidefinite Program?

- $\mathcal{S}^m := \mathcal{S}^m(\mathbb{R})$ space of all $m \times m$ symmetric matrices (real entries)
- $A \in \mathcal{S}^m$ is *positive semidefinite* (PSD), denoted $A \succeq 0$ if
  1. all eigenvalues of A are *non-negative*
  2. $A = Y^T Y$ for some $Y \in \mathbb{R}^{d \times m}$ where $d \leq m$
  3. $z^T A z \geq 0$ for any $z \in \mathbb{R}^m$
  4. and more...

Semidefinite Programming deals with problems of the form

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & x_1 \cdot A_1 + \cdots + x_n \cdot A_n \succeq B \\ & x \in \mathbb{R}^n \end{aligned}$$

What is a Semidefinite Program?

- $\mathcal{S}^m := \mathcal{S}^m(\mathbb{R})$ space of all $m \times m$ symmetric matrices (real entries)
- $A \in \mathcal{S}^m$ is *positive semidefinite* (PSD), denoted $A \succeq 0$ if
  1. all eigenvalues of A are *non-negative*
  2. $A = Y^T Y$ for some $Y \in \mathbb{R}^{d \times m}$ where $d \leq m$
  3. $z^T A z \geq 0$ for any $z \in \mathbb{R}^m$
  4. and more...

Semidefinite Programming deals with problems of the form

$$\begin{array}{rl} \text{minimize} & c^T x \\ \text{subject to} & x_1 \cdot A_1 + \cdots + x_n \cdot A_n \succeq B \\ & x \in \mathbb{R}^n \end{array}$$

Where we use $C \succeq D$ to denote that $C - D \succeq 0$ (i.e., $C - D$ is PSD).

How does it generalize Linear Programming?

Linear Programming

$$\begin{aligned} \text{minimize} \quad & a^T x \\ \text{subject to} \quad & Cx \geq b \\ & x \in \mathbb{R}^n \end{aligned}$$

How does it generalize Linear Programming?

Linear Programming

$$\begin{array}{rl} \text{minimize} & a^T x \\ \text{subject to} & Cx \geq b \\ & x \in \mathbb{R}^n \end{array}$$

Semidefinite Programming

$$\begin{array}{rl} \text{minimize} & c^T x \\ \text{subject to} & x_1 \cdot A_1 + \cdots + x_n \cdot A_n \succeq B \\ & x \in \mathbb{R}^n \end{array}$$

How does it generalize Linear Programming?

Linear Programming → **Semidefinite Programming**

minimize $a^T x$	minimize $c^T x$
subject to $Cx \geq b$	subject to $x_1 \cdot A_1 + \cdots + x_n \cdot A_n \succeq B$
$x \in \mathbb{R}^n$	$x \in \mathbb{R}^n$

$\begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$

Set A_i's to be diagonal matrices, and $B = diag(b_1, \ldots, b_m)$

$$\sum_{i=1}^{n} C_{ki} x_i \geq b_k$$

$(A_i)_{kk} = C_{ki}$ — k^{th} diagonal entry

$$x_1 \begin{pmatrix} C_{11} & & 0 \\ & C_{21} & \\ 0 & & \ddots \\ & & & C_{m1} \end{pmatrix} + \cdots + x_n \begin{pmatrix} C_{1n} & & 0 \\ & C_{2n} & \\ 0 & & \ddots \\ & & & C_{mn} \end{pmatrix} \succeq \begin{pmatrix} b_1 & & 0 \\ & \ddots & \\ 0 & & b_m \end{pmatrix}$$

$$\Longleftrightarrow \sum_{i=1}^{n} C_{ki} x_i \geq b_k$$

Why should I care?

- Linear Programs appear everywhere in life: many problems of interest (resource allocation problems) can be modelled as linear program!

Why should I care?

- Linear Programs appear everywhere in life: many problems of interest (resource allocation problems) can be modelled as linear program!
- Semidefinite Programming is no different!

Why should I care?

- Linear Programs appear everywhere in life: many problems of interest (resource allocation problems) can be modelled as linear program!
- Semidefinite Programming is no different!
  - equilibrium analysis of dynamics and control (flight controls, robotics, etc.)
    TODAY
  - robust optimization
  - statistics and ML
  - continuous games
  - software verification
  - filter design
  - quantum computation and information
  - automated theorem proving
  - packing problems
  - many more

Why should I care?

- Linear Programs appear everywhere in life: many problems of interest (resource allocation problems) can be modelled as linear program!
- Semidefinite Programming is no different!
  - equilibrium analysis of dynamics and control (flight controls, robotics, etc.)
  - robust optimization
  - statistics and ML
  - continuous games
  - software verification
  - filter design
  - quantum computation and information
  - automated theorem proving
  - packing problems
  - many more
- See more here

`https://windowsontheory.org/2016/08/27/proofs-beliefs-and-algorithms-through-the-lens-of-sum-of-squares/`

Important Questions

$$
\begin{aligned}
\text{minimize} \quad & c^T x \\
\text{subject to} \quad & x_1 \cdot A_1 + \cdots + x_n \cdot A_n \succeq B \\
& x \in \mathbb{R}^n
\end{aligned}
$$

Important Questions

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & x_1 \cdot A_1 + \cdots + x_n \cdot A_n \succeq B \\ & x \in \mathbb{R}^n \end{aligned}$$

1. When is a Semidefinite Program *feasible*?
   - Is there a solution to the constraints at all?

Important Questions

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & x_1 \cdot A_1 + \cdots + x_n \cdot A_n \succeq B \\ & x \in \mathbb{R}^n \end{aligned}$$

1. When is a Semidefinite Program *feasible*?
   - Is there a solution to the constraints at all?
2. When is a Semidefinite Program *bounded*?
   - Is there a minimum? Is the minimum achievable? Or is the minimum $-\infty$?

Important Questions

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & x_1 \cdot A_1 + \cdots + x_n \cdot A_n \succeq B \\ & x \in \mathbb{R}^n \end{aligned}$$

1. When is a Semidefinite Program *feasible*?
   - Is there a solution to the constraints at all?
2. When is a Semidefinite Program *bounded*?
   - Is there a minimum? Is the minimum achievable? Or is the minimum $-\infty$?
3. Can we characterize *optimality*?
   - How can we know that we found a minimum solution?
   - Do these solutions have nice description?
   - Do the solutions have *small bit complexity*?

Important Questions

$$
\begin{aligned}
\text{minimize} \quad & c^T x \\
\text{subject to} \quad & x_1 \cdot A_1 + \cdots + x_n \cdot A_n \succeq B \\
& x \in \mathbb{R}^n
\end{aligned}
$$

1. When is a Semidefinite Program *feasible*?
   - Is there a solution to the constraints at all?
2. When is a Semidefinite Program *bounded*?
   - Is there a minimum? Is the minimum achievable? Or is the minimum $-\infty$?
3. Can we characterize *optimality*?
   - How can we know that we found a minimum solution?
   - Do these solutions have nice description?
   - Do the solutions have *small bit complexity*?
4. How do we design *efficient algorithms* that find *optimal solutions* to Semidefinite Programs?

Spectrahedra

To understand SDPs, we need to understand their *feasible regions*, which are called *spectrahedra* and described as *Linear Matrix Inequalities* (LMIs).

Spectrahedra

To understand SDPs, we need to understand their *feasible regions*, which are called *spectrahedra* and described as *Linear Matrix Inequalities* (LMIs).

Definition (Linear Matrix Inequalities)

A linear matrix inequality is an inequality of the form:

$$A_0 + \sum_{i=1}^{n} A_i x_i \succeq 0,$$

where $A_0, \ldots, A_n$ are *symmetric matrices*.

Spectrahedra

To understand SDPs, we need to understand their *feasible regions*, which are called *spectrahedra* and described as *Linear Matrix Inequalities* (LMIs).

Definition (Linear Matrix Inequalities)

A linear matrix inequality is an inequality of the form:

$$A_0 + \sum_{i=1}^{n} A_i x_i \succeq 0,$$

where $A_0, \ldots, A_n$ are *symmetric matrices*.

Definition (Spectrahedron)

A spectrahedron is a set defined by finitely many LMIs. In other words, it can be defined as:

$$S = \left\{ x \in \mathbb{R}^n \;\middle|\; \sum_{i=1}^{n} A_i x_i \succeq B, \quad A_i, B \in \mathcal{S}^m \right\}$$

Spectrahedra

To understand SDPs, we need to understand their *feasible regions*, which are called *spectrahedra* and described as *Linear Matrix Inequalities* (LMIs).

If S defined by $\sum_{i=1}^{n} A_i x_i \succeq B_1$ and $\sum_{i=1}^{n} C_i x_i \succeq B_2$ then S defined by

$$\sum_{i=1}^{n} x_i \begin{pmatrix} A_i & 0 \\ 0 & C_i \end{pmatrix} \succeq \begin{pmatrix} B_1 & 0 \\ 0 & B_2 \end{pmatrix}.$$

2x2 block

Definition (Spectrahedron)

A spectrahedron is a set defined by finitely many LMIs. In other words, it can be defined as:

$$S = \left\{ x \in \mathbb{R}^n \;\middle|\; \sum_{i=1}^{n} A_i x_i \succeq B, \quad A_i, B \in \mathcal{S}^m \right\}$$

Spectrahedra

$A, B \succeq 0 \quad z^T(A+B)z = \underbrace{z^TAz}_{\geq 0} + \underbrace{z^TBz}_{\geq 0} \geq 0$

To understand SDPs, we need to understand their *feasible regions*, which are called *spectrahedra* and described as *Linear Matrix Inequalities* (LMIs).

Spectrahedra are convex: $z = \alpha x + (1-\alpha) y$

$x, y \in S$ then $\alpha \in [0,1]$ we have

$$\sum_{i=1}^{n} A_i \underbrace{(\alpha x_i + (1-\alpha) y_i)}_{z_i} = \underbrace{\alpha}_{\geq 0} \underbrace{\sum_{i=1}^{n} A_i x_i}_{\succeq B} + \underbrace{(1-\alpha)}_{\geq 0} \underbrace{\sum_{i=1}^{n} A_i y_i}_{\succeq (1-\alpha)B} \succeq$$

$$\alpha B + (1-\alpha) B = B \quad \therefore \alpha x + (1-\alpha) y \in S.$$

Definition (Spectrahedron)

A spectrahedron is a set defined by finitely many LMIs. In other words, it can be defined as:

$$S = \left\{ x \in \mathbb{R}^n \;\middle|\; \sum_{i=1}^{n} A_i x_i \succeq B, \quad A_i, B \in \mathcal{S}^m \right\}$$

Example of Spectrahedron

Polyhedron:

$P = \{ x \in \mathbb{R}^n \mid Ax \geq b \}$ LP

$\sum_{i=1}^{n} A_{ki} x_i \geq b_k$ k^{th} constraint

$$\sum_{i=1}^{n} x_i \begin{pmatrix} A_{1i} & & \\ & \ddots & \\ & & A_{mi} \end{pmatrix} \succeq \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_m \end{pmatrix}$$

LMI

SDP

Example of Spectrahedron

Circle:

$$\mathcal{C} = \{(x,y) \in \mathbb{R}^2 \mid x^2+y^2 \leq 1\}$$

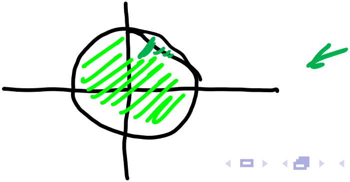
$$\mathcal{C} = \left\{(x,y) \in \mathbb{R}^2 \;\middle|\; \underbrace{\begin{pmatrix} 1+x & y \\ y & 1-x \end{pmatrix}}_{Z} \succeq 0 \right\}$$

PSD

$$1+x \succeq 0$$
$$1-x \succeq 0$$

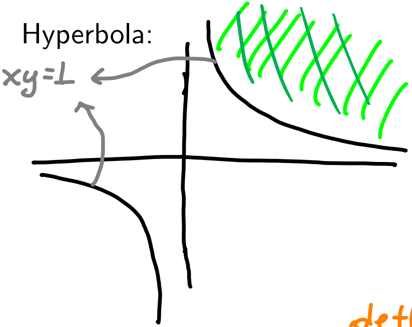
$$\left.\begin{array}{l} \det(Z) \succeq 0 \\ (1+x)(1-x) - y^2 \end{array}\right\} \Rightarrow \begin{array}{l} -1 \leq x \leq 1 \\ 1-x^2-y^2 \succeq 0 \end{array}$$

$$\Rightarrow \begin{array}{l} -1 \leq x \leq 1 \\ x^2+y^2 \leq 1 \end{array}$$



Example of Spectrahedron

Hyperbola:



$$\mathcal{H} = \left\{ (x,y) \in \mathbb{R}^2 \;\middle|\; \begin{matrix} x, y \geq 0 \\ xy \geq 1 \end{matrix} \right\}$$

$$\mathcal{H} = \left\{ (x,y) \in \mathbb{R}^2 \;\middle|\; \begin{pmatrix} x & 1 \\ 1 & y \end{pmatrix} \succeq 0 \right\}$$

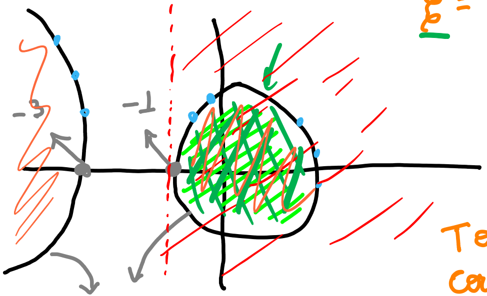
$$\left.\begin{matrix} x \geq 0 \\ y \geq 0 \\ \det\begin{pmatrix} x & 1 \\ 1 & y \end{pmatrix} \geq 0 \end{matrix}\right\} \Rightarrow \begin{matrix} x, y \geq 0 \\ xy \geq 1 \end{matrix}$$

$xy - 1$

Example of Spectrahedron

$x+1 \geq 0$

Elliptic curve: (oval part)



$$0 = -2y^2 - x^3 - 3x^2 + x + 3$$

$\det(A(x,y))$

$$\mathcal{E} = \left\{ (x,y) \in \mathbb{R}^2 \;\middle|\; A(x,y) = \begin{bmatrix} x+1 & 0 & y \\ 0 & 2 & -x-1 \\ y & -x-1 & 2 \end{bmatrix} \succeq 0 \right\}$$

To see that green region corresponds to $\mathcal{E}$ need to show that oval corresponds to being in $\mathcal{E}$:

$(t-\lambda_1)(t-\lambda_2)(t-\lambda_3) = t^3 - (\lambda_1+\lambda_2+\lambda_3)t^2$

$A(x,y) \succeq 0$ iff $\det(tI - A(x,y))$ has only ≥ 0 roots

$$\det(tI - A(x,y)) = t^3 - (x+5)t^2 + (-x^2+2x-y^2+7)t - \det(A(x,y))$$

≥ 0 roots $\Leftrightarrow$ $x+5 \geq 0$, $-x^2+2x-y^2+7 \geq 0$, $\det(A(x,y)) \geq 0$

extra inequalities isolate component

Projected Spectrahedron

For both LPs and SDPs, it is enough to obtain a *linear projection of spectrahedron* (or polyhedron, if in LP).

LP projections of polyhedra
are polyhedra

Projected Spectrahedron

For both LPs and SDPs, it is enough to obtain a *linear projection of spectrahedron* (or polyhedron, if in LP).

Definition (Projected Spectrahedron)

A set $S \in \mathbb{R}^n$ is a *projected spectrahedron* if it has the form:

$$S = \left\{ x \in \mathbb{R}^n \mid \exists y \in \mathbb{R}^t \text{ s.t. } \sum_{i=1}^{n} A_i x_i + \sum_{j=1}^{t} B_j y_j \succeq C, \quad A_i, B_j, C \in \mathcal{S}^m \right\}$$

S projection of:

$$T := \left\{ (x, y) \in \mathbb{R}^{n+t} \;\middle|\; \sum_i A_i x_i + \sum_j B_j y_j \succeq C \right\}$$

S projection of T to firs n coordinates

Projected Spectrahedron

For both LPs and SDPs, it is enough to obtain a *linear projection of spectrahedron* (or polyhedron, if in LP).

Definition (Projected Spectrahedron)

A set $S \in \mathbb{R}^n$ is a *projected spectrahedron* if it has the form:

$$S = \left\{ x \in \mathbb{R}^n \;\middle|\; \exists y \in \mathbb{R}^t \text{ s.t. } \sum_{i=1}^{n} A_i x_i + \sum_{j=1}^{t} B_j y_j \succeq C, \quad A_i, B_j, C \in \mathcal{S}^m \right\}$$

minimize $c^T x$
s.t. $x \in S$

$\Leftarrow\!\Rightarrow$

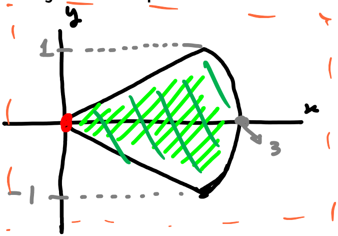
minimize $c^T x$
s.t. $(x, y) \in T$

$c^T x = (c^T, 0)\begin{pmatrix} x \\ y \end{pmatrix}$, where $(c^T, 0) = \hat{c}$

SDP

Example of ~~Spectrahedron~~ Projection of Spectrahedron

Projection quadratic cone intersected with halfspace:



$$S = \left\{ (x,y) \in \mathbb{R}^2 \;\middle|\; \exists z \in \mathbb{R} \text{ s.t. } A = \begin{pmatrix} z+y & 2z-x \\ 2z-x & z-y \end{pmatrix} \succeq 0,\; z \le 1 \right\}$$

$\det(A) \geq 0$

In $\mathbb{R}^3$, (x,y,z) would be given by

$z \le 1$ intersect with cone $z^2 \geq y^2 + (2z-x)^2$

and equations $z+y \geq 0$, $z-y \geq 0$ $(\therefore\ z \geq 0)$

(convince yourself that we will also have $x \geq 0$)

Remark: unlike polyhedral case, projection of spectrahedra MAY NOT be spectrahedra (this is an example)

How do we test membership in the Spectrahedron?

- To be able to optimize, we must be able to test whether a given point $x \in \mathbb{R}^n$ is inside our spectrahedron

$$S = \left\{ x \in \mathbb{R}^n \;\middle|\; \sum_{i=1}^{n} A_i x_i \succeq B, \quad A_i, B \in \mathcal{S}^m \right\}.$$

$x \in S$?

How do we test membership in the Spectrahedron?

- To be able to optimize, we must be able to test whether a given point $x \in \mathbb{R}^n$ is inside our spectrahedron

$$S = \left\{ x \in \mathbb{R}^n \;\middle|\; \sum_{i=1}^{n} A_i x_i \succeq B, \quad A_i, B \in \mathcal{S}^m \right\}.$$

- Note that $x \in S$ is (by definition) equivalent to

$$Z = \sum_{i=1}^{n} A_i x_i - B \succeq 0$$

How do we test membership in the Spectrahedron?

- To be able to optimize, we must be able to test whether a given point $x \in \mathbb{R}^n$ is inside our spectrahedron

$$S = \left\{ x \in \mathbb{R}^n \;\middle|\; \sum_{i=1}^{n} A_i x_i \succeq B, \quad A_i, B \in \mathcal{S}^m \right\}.$$

- Note that $x \in S$ is (by definition) equivalent to

$$Z = \sum_{i=1}^{n} A_i x_i - B \succeq 0$$

- So, how do we efficiently check if $Z \succeq 0$?

How do we test membership in the Spectrahedron?

- To be able to optimize, we must be able to test whether a given point $x \in \mathbb{R}^n$ is inside our spectrahedron

$$S = \left\{ x \in \mathbb{R}^n \;\middle|\; \sum_{i=1}^{n} A_i x_i \succeq B, \quad A_i, B \in \mathcal{S}^m \right\}.$$

- Note that $x \in S$ is (by definition) equivalent to

$$Z = \sum_{i=1}^{n} A_i x_i - B \succeq 0$$

- So, how do we efficiently check if $Z \succeq 0$?
- Symmetric Gaussian Elimination!

How do we test membership in the Spectrahedron?

- To be able to optimize, we must be able to test whether a given point $x \in \mathbb{R}^n$ is inside our spectrahedron

$$S = \left\{ x \in \mathbb{R}^n \;\middle|\; \sum_{i=1}^{n} A_i x_i \succeq B, \quad A_i, B \in \mathcal{S}^m \right\}.$$

- Note that $x \in S$ is (by definition) equivalent to

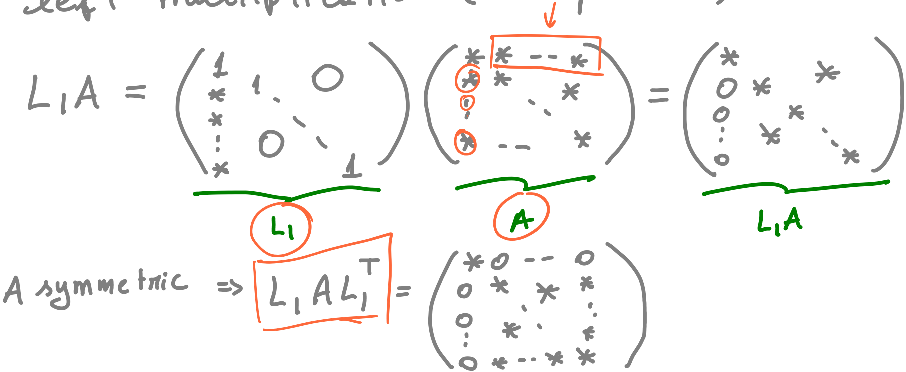
$$Z = \sum_{i=1}^{n} A_i x_i - B \succeq 0$$

- So, how do we efficiently check if $Z \succeq 0$?
- Symmetric Gaussian Elimination!
- We will use following characterizations of PSDness of symmetric $A \in \mathcal{S}^m$
  1. all eigenvalues of A are *non-negative*
  2. $A = LDL^T$ for some L lower triangular and unit diagonal, D diagonal and non-negative
  3. $z^T A z \geq 0$ for any $z \in \mathbb{R}^m$
  4. Any principal minor of A has non-negative determinant

How do we test membership in the Spectrahedron?

- **Input:** symmetric matrix $A \in \mathcal{S}^m$
- **Output:** YES if $A \succeq 0$, NO otherwise (and output $z \in \mathbb{R}^m$ such that $z^T A z < 0$)

Clear out first column of A by left multiplication (row operations)

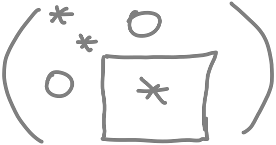


How do we test membership in the Spectrahedron?

- **Input:** symmetric matrix $A \in \mathcal{S}^m$
- **Output:** YES if $A \succeq 0$, NO otherwise (and output $z \in \mathbb{R}^m$ such that $z^T A z < 0$)

Similarly, clear out second row and column

$L_2 L_1 A L_1^T L_2^T =$



and so on …

How do we test membership in the Spectrahedron?

- **Input:** symmetric matrix $A \in \mathcal{S}^m$
- **Output:** YES if $A \succeq 0$, NO otherwise (and output $z \in \mathbb{R}^m$ such that $z^T A z < 0$)

Our algorithm ends when

$$\underbrace{L_m L_{m-1} \cdots L_2 L_1}_{L} A \underbrace{L_1^T L_2^T \cdots L_m^T}_{L^T} = \underbrace{\begin{pmatrix} * & & 0 \\ & * & \\ 0 & & \ddots \\ & & & * \end{pmatrix}}_{D \text{ diagonal}}$$

$$LAL^T = D$$

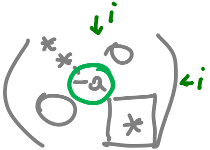
Product of L_i's still same shape $\begin{pmatrix} 1 & & 0 \\ & \ddots & \\ * & & 1 \end{pmatrix}$.

or, our algorithm halts if the following happen:

How do we test membership in the Spectrahedron?

- **Input:** symmetric matrix $A \in \mathcal{S}^m$
- **Output:** YES if $A \succeq 0$, NO otherwise (and output $z \in \mathbb{R}^m$ such that $z^T A z < 0$)

If at any point we have $LAL^T =$

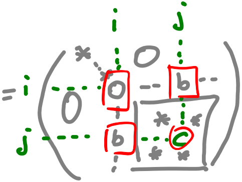


with $a > 0$ then return NO

and $z = L^T e_i$

$z^T A z = e_i^T (LAL^T) e_i = -a < 0$.

If at any point we have $LAL^T =$



with $b \neq 0$ then return NO

Practice problem: what is z here?

How do we test membership in the Spectrahedron?

- **Input:** symmetric matrix $A \in \mathcal{S}^m$
- **Output:** YES if $A \succeq 0$, NO otherwise (and output $z \in \mathbb{R}^m$ such that $z^T A z < 0$)

If our Symmetric Gaussian Elimination runs until the end, we have that $LAL^T = D$ with $D_{ii} \geq 0$ $\forall i \in [m]$, all other entries zero $\therefore D \succeq 0 \Rightarrow A \succeq 0$.

- Our algorithm runs in time strongly polynomial.

iterations depends on combinatorial size of problem

Working with Symmetric Matrices

Definition (Frobenius Inner Product)

$A, B \in \mathcal{S}^m$, define the *Frobenius inner product* as

$$\langle A, B \rangle := \mathrm{tr}[AB] = \sum_{i,j} A_{ij} B_{ij}$$

- This is the "usual inner product" if you think of the matrices as vectors
- Thus, have the norm

$$\|A\|_F = \sqrt{\langle A, A \rangle} = \sqrt{\sum_{i,j} A_{ij}^2}$$

- With this norm, can talk about the *polar dual* to a given spectrahedron $S \subseteq \mathcal{S}^m$:

$$S^\circ = \{Y \in \mathcal{S}^m \mid \langle Y, X \rangle \leq 1, \ \forall X \in S\}$$

Standard Primal Form

Just like in Linear Programming, we can represent SDPs in standard form:

$$\begin{aligned} \text{minimize} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i \\ & \boxed{X \succeq 0} \end{aligned}$$

Variable constraints

Where now:

Standard Primal Form

Just like in Linear Programming, we can represent SDPs in standard form:

$$\begin{aligned} \text{minimize} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i \\ & X \succeq 0 \end{aligned}$$

Where now:

- the variables are encoded in a positive semidefinite matrix X,

Standard Primal Form

Just like in Linear Programming, we can represent SDPs in standard form:

$$\begin{aligned} \text{minimize} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i \\ & X \succeq 0 \end{aligned}$$

Where now:

- the variables are encoded in a positive semidefinite matrix X,
- each constraint is given by an inner product $\langle A_i, X \rangle = b_i$

Standard Primal Form

Just like in Linear Programming, we can represent SDPs in standard form:

$$\begin{aligned} \text{minimize} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i \\ & X \succeq 0 \end{aligned}$$

$c^T x = \langle c, x \rangle$

$Ax = b$

$x \geq 0$

Where now:

- the variables are encoded in a positive semidefinite matrix X,
- each constraint is given by an inner product $\langle A_i, X \rangle = b_i$
- Note the similarity with LP standard primal. Can obtain LP standard form by making X and A_i's to be diagonal

Standard Primal Form

Just like in Linear Programming, we can represent SDPs in standard form:

$$\begin{array}{rl} \text{minimize} & \langle C, X\rangle \\ \text{subject to} & \langle A_i, X\rangle = b_i \\ & X \succeq 0 \end{array}$$

Where now:

- the variables are encoded in a positive semidefinite matrix X,
- each constraint is given by an inner product $\langle A_i, X\rangle = b_i$
- Note the similarity with LP standard primal. Can obtain LP standard form by making X and A_i's to be diagonal
- How is that an LMI though?

Standard Primal Form

$$\begin{array}{ll} \text{minimize} & \langle C, X\rangle \\ \text{subject to} & \langle A_i, X\rangle = b_i \\ & X \succeq 0 \end{array}$$

$\langle A_i, X\rangle = b_i \quad \sum_{k,l} (A_i)_{kl}\, x_{kl} = b_i \quad \Rightarrow$

$$\sum_{k,l} x_{kl} \begin{pmatrix} (A_1)_{kl} & & \\ & \ddots & \\ & & (A_t)_{kl} \end{pmatrix} = \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_t \end{pmatrix}$$

$$X \succeq 0$$

are the two LMIs defining our set.

Example

Value: $1-\sqrt{2}$

$$\mathrm{OPT} = \begin{bmatrix} \frac{2-\sqrt{2}}{4} & -\frac{1}{2\sqrt{2}} \\ -\frac{1}{2\sqrt{2}} & \frac{2+\sqrt{2}}{4} \end{bmatrix}$$

NOT Rational

Primal Program

$$\begin{aligned} \text{minimize} \quad & 2x_{11} + 2x_{12} \\ \text{subject to} \quad & x_{11} + x_{22} = 1 \\ & \begin{pmatrix} x_{11} & x_{12} \\ x_{12} & x_{22} \end{pmatrix} \succeq 0 \end{aligned}$$

$$A_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$b_1 = 1$$

$$C = \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}$$

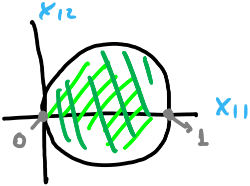
Constraints satisfied iff $x_{11} x_{22} \geq x_{12}^2$ ($x_{11} \geq 0$, $x_{22} \geq 0$)

$$\Leftrightarrow x_{11}(1-x_{11}) \geq x_{12}^2$$

$$\Leftrightarrow 1 - 1 + 4x_{11} - 4x_{11}^2 \geq 4x_{12}^2$$

$$\Leftrightarrow 1 \geq (2x_{12})^2 + (2x_{11}-1)^2$$

Feasible set: closed disk



Semidefinite Programming Duality

Consider our SDP:

$$\begin{aligned} \text{minimize} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i \\ & X \succeq 0 \end{aligned}$$

Semidefinite Programming Duality

Consider our SDP:

$$\begin{aligned} \text{minimize} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i \\ & X \succeq 0 \end{aligned}$$

Semidefinite Programming Duality

Consider our SDP:

$$\begin{aligned} \text{minimize} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i \\ & X \succeq 0 \end{aligned}$$

- If we look at what happens when we multiply i^{th} equality by a variable y_i:

$$\sum_{i=1}^{t} y_i \cdot \langle A_i, X \rangle = \sum_{i=1}^{t} y_i \cdot b_i \quad \Rightarrow \quad \left\langle \sum_{i=1}^{t} y_i A_i \ , \ X \right\rangle = y^T b$$

Semidefinite Programming Duality

Consider our SDP:

$A \preceq B \Leftrightarrow \langle A, X\rangle \leq \langle B, X\rangle$

$$\begin{aligned} \text{minimize} \quad & \langle C, X\rangle \\ \text{subject to} \quad & \langle A_i, X\rangle = b_i \\ & X \succeq 0 \end{aligned}$$

- If we look at what happens when we multiply i^{th} equality by a variable y_i:

$$\sum_{i=1}^{t} y_i \cdot \langle A_i, X\rangle = \sum_{i=1}^{t} y_i \cdot b_i \quad \Rightarrow \quad \left\langle \sum_{i=1}^{t} y_i A_i \ , \ X \right\rangle = y^T b$$

$\preceq C$

- Thus, if $\sum_{i=1}^{t} y_i A_i \preceq C$, then we have:

$$y^T b = \left\langle \sum_{i=1}^{t} y_i A_i \ , \ X \right\rangle \leq \langle C, X\rangle$$

Semidefinite Programming Duality

Consider our SDP:

$$\begin{aligned} \text{minimize} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i \\ & X \succeq 0 \end{aligned}$$

- If we look at what happens when we multiply i^{th} equality by a variable y_i:

$$\sum_{i=1}^{t} y_i \cdot \langle A_i, X \rangle = \sum_{i=1}^{t} y_i \cdot b_i \quad \Rightarrow \quad \left\langle \sum_{i=1}^{t} y_i A_i \ , \ X \right\rangle = y^T b$$

- Thus, if $\sum_{i=1}^{t} y_i A_i \preceq C$, then we have:

lower bd   value primal

$$y^T b = \left\langle \sum_{i=1}^{t} y_i A_i \ , \ X \right\rangle \leq \langle C, X \rangle$$

- $y^T b$ is a *lower bound* on the solution to our SDP!

Semidefinite Programming Duality

Consider the following SDPs:

Primal SDP

$$\begin{aligned} \text{minimize} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i \\ & X \succeq 0 \end{aligned}$$

Dual SDP

$$\begin{aligned} \text{maximize} \quad & y^T b \\ \text{subject to} \quad & \sum_{i=1}^{t} y_i A_i \preceq C \end{aligned}$$

lower bound

Semidefinite Programming Duality

Consider the following SDPs:

Primal SDP		*Dual SDP*	
minimize	$\langle C, X \rangle$	maximize	$y^T b$
subject to	$\langle A_i, X \rangle = b_i$	subject to	$\sum_{i=1}^{t} y_i A_i \preceq C$
	$X \succeq 0$		

- From previous slide

$$\sum_{i=1}^{t} y_i A_i \preceq C \Rightarrow y^T b \text{ is a lower bound on value of Primal}$$

Semidefinite Programming Duality

Consider the following SDPs:

Primal SDP

$$\begin{aligned} \text{minimize} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i \\ & X \succeq 0 \end{aligned}$$

Dual SDP

$$\begin{aligned} \text{maximize} \quad & y^T b \\ \text{subject to} \quad & \sum_{i=1}^{t} y_i A_i \preceq C \end{aligned}$$

- From previous slide

$$\sum_{i=1}^{t} y_i A_i \preceq C \Rightarrow y^T b \text{ is a lower bound on value of Primal}$$

- Thus, the optimal (maximum) value of *dual LP* lower bounds the optimal (minimum) value of the *Primal LP*!

Semidefinite Programming Duality

Consider the following SDPs:

Primal SDP	*Dual SDP*
minimize $\langle C, X\rangle$ subject to $\langle A_i, X\rangle = b_i$ $X \succeq 0$	maximize $y^T b$ subject to $\sum_{i=1}^{t} y_i A_i \preceq C$

- From previous slide

$$\sum_{i=1}^{t} y_i A_i \preceq C \Rightarrow y^T b \text{ is a lower bound on value of Primal}$$

- Thus, the optimal (maximum) value of *dual LP* lower bounds the optimal (minimum) value of the *Primal LP*!

Theorem (Weak Duality)

Let X be a feasible solution of the primal SDP and y be a feasible solution of the dual SDP. Then

$$y^T b \leq \langle C, X\rangle.$$

Remarks on Duality

Primal SDP

$$\begin{aligned} \text{minimize} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i \\ & X \succeq 0 \end{aligned}$$

Dual SDP

$$\begin{aligned} \text{maximize} \quad & y^T b \\ \text{subject to} \quad & \sum_{i=1}^{t} y_i A_i \preceq C \end{aligned}$$

Remarks on Duality

Primal SDP

$$\begin{aligned} \text{minimize} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i \\ & X \succeq 0 \end{aligned}$$

Dual SDP

$$\begin{aligned} \text{maximize} \quad & y^T b \\ \text{subject to} \quad & \sum_{i=1}^{t} y_i A_i \preceq C \end{aligned}$$

Theorem (Complementary Slackness)

Let X be a feasible solution of the primal SDP and y be a feasible solution of the dual SDP. If (X, y) satisfy the complementary slackness condition

$$\left(C - \sum_{i=1}^{t} y_i A_i \right) X = 0$$

Then (X, y) are primal and dual optimum solutions of the SDP problem.

Remarks on Duality

Primal SDP

$$\begin{aligned} \text{minimize} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i \\ & X \succeq 0 \end{aligned}$$

Dual SDP

$$\begin{aligned} \text{maximize} \quad & y^T b \\ \text{subject to} \quad & \sum_{i=1}^{t} y_i A_i \preceq C \end{aligned}$$

Theorem (Complementary Slackness)

Let X be a feasible solution of the primal SDP and y be a feasible solution of the dual SDP. If (X, y) satisfy the complementary slackness condition

$$\left(C - \sum_{i=1}^{t} y_i A_i \right) X = 0$$

Then (X, y) are primal and dual optimum solutions of the SDP problem.

Complementary slackness gives us *sufficient* conditions to check optimality of our solutions.

Strong Duality

Primal SDP

$$\begin{aligned} \text{minimize} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i \\ & X \succeq 0 \end{aligned}$$

Dual SDP

$$\begin{aligned} \text{maximize} \quad & y^T b \\ \text{subject to} \quad & \sum_{i=1}^{t} y_i A_i \preceq C \end{aligned}$$

Strong Duality

Primal SDP

$$\begin{aligned} \text{minimize} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i \\ & X \succeq 0 \end{aligned}$$

Dual SDP

$$\begin{aligned} \text{maximize} \quad & y^T b \\ \text{subject to} \quad & \sum_{i=1}^{t} y_i A_i \preceq C \end{aligned}$$

- Strong duality in SDPs is a bit more complex than in LPs.

Strong Duality

Primal SDP

$$\begin{aligned} \text{minimize} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i \\ & X \succeq 0 \end{aligned}$$

Dual SDP

$$\begin{aligned} \text{maximize} \quad & y^T b \\ \text{subject to} \quad & \sum_{i=1}^{t} y_i A_i \preceq C \end{aligned}$$

- Strong duality in SDPs is a bit more complex than in LPs.
- Both primal and dual may be feasible, and yet strong duality may not hold! (you will see this in Homework)

Strong Duality

Primal SDP		*Dual SDP*	
minimize	$\langle C, X \rangle$	maximize	$y^T b$
subject to	$\langle A_i, X \rangle = b_i$	subject to	$\sum_{i=1}^{t} y_i A_i \preceq C$
	$X \succeq 0$		

- Strong duality in SDPs is a bit more complex than in LPs.
- Both primal and dual may be feasible, and yet strong duality may not hold!
- But under mild conditions, strong duality holds!

Strong Duality

Primal SDP

$$\begin{array}{ll} \text{minimize} & \langle C, X \rangle \\ \text{subject to} & \langle A_i, X \rangle = b_i \\ & X \succeq 0 \end{array}$$

Dual SDP

$$\begin{array}{ll} \text{maximize} & y^T b \\ \text{subject to} & \sum_{i=1}^{t} y_i A_i \preceq C \end{array}$$

- Strong duality in SDPs is a bit more complex than in LPs.
- Both primal and dual may be feasible, and yet strong duality may not hold!
- But under mild conditions, strong duality holds!
- Primal SDP is *strictly feasible* if there is feasible solution $X \succ 0$.
- Dual SDP is *strictly feasible* if there is feasible $\sum_{i=1}^{t} y_i A_i \prec C$.

Slater conditions

Strong Duality

Primal SDP

$$\begin{aligned} \text{minimize} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i \\ & X \succeq 0 \end{aligned}$$

Dual SDP

$$\begin{aligned} \text{maximize} \quad & y^T b \\ \text{subject to} \quad & \sum_{i=1}^{t} y_i A_i \preceq C \end{aligned}$$

- Strong duality in SDPs is a bit more complex than in LPs.
- Both primal and dual may be feasible, and yet strong duality may not hold!
- But under mild conditions, strong duality holds!
- Primal SDP is *strictly feasible* if there is feasible solution $X \succ 0$.
- Dual SDP is *strictly feasible* if there is feasible $\sum_{i=1}^{t} y_i A_i \prec C$.

Theorem (Strong Duality under Slater Conditions)

If primal SDP and dual SDP are both strictly feasible, then

$$\textit{max dual} = \textit{min of primal.}$$

- Part I
  - Why Semidefinite Programming?
  - Convex Algebraic Geometry
- Part II
  - Duality Theory
  - Application: Control Theory
- Conclusion
- Acknowledgements

Stability of Linear Systems

Setup:

- Linear difference equation

$$x(t+1) = Ax(t), \quad x(0) = x_0$$

- Discrete-time dynamical system.[1]

[1] When A non-negative and x_0 non-negative we have Markov chains.

Stability of Linear Systems

Setup:

- Linear difference equation

$$x(t+1) = Ax(t), \quad x(0) = x_0$$

- Discrete-time dynamical system.[1]
- Used to model time evolution of

[1] When A non-negative and x_0 non-negative we have Markov chains.

Stability of Linear Systems

Setup:

- Linear difference equation

$$x(t+1) = Ax(t), \quad x(0) = x_0$$

- Discrete-time dynamical system.[1]
- Used to model time evolution of
  - Temperatures of objects
  - Size of population
  - Voltage of electrical circuits
  - Concentration of chemical mixtures

[1] When A non-negative and x_0 non-negative we have Markov chains.

Stability of Linear Systems

Setup:

- Linear difference equation

$$x(t+1) = Ax(t), \quad x(0) = x_0$$

- Discrete-time dynamical system.[1]
- Used to model time evolution of
  - Temperatures of objects
  - Size of population
  - Voltage of electrical circuits
  - Concentration of chemical mixtures
- **Question:** when $t \to \infty$, under what conditions will $x(t)$ remain bounded? Or go to zero?

[1]When A non-negative and x_0 non-negative we have Markov chains.

Stability of Linear Systems

Setup:

- Linear difference equation

$$x(t+1) = Ax(t), \quad x(0) = x_0$$

- Discrete-time dynamical system.[1]
- Used to model time evolution of
  - Temperatures of objects
  - Size of population
  - Voltage of electrical circuits
  - Concentration of chemical mixtures
- **Question:** when $k \to \infty$, under what conditions will $x(t)$ remain bounded? Or go to zero?
- When system converges to zero, we say it is *stable*.

[1] When A non-negative and x_0 non-negative we have Markov chains.

Stability of Linear Systems

Setup:

- Linear difference equation

$$x(t+1) = Ax(t), \quad x(0) = x_0$$

- Discrete-time dynamical system.[1]
- Used to model time evolution of
  - Temperatures of objects
  - Size of population
  - Voltage of electrical circuits
  - Concentration of chemical mixtures
- **Question:** when $k \to \infty$, under what conditions will $x(t)$ remain bounded? Or go to zero?
- When system converges to zero, we say it is *stable*.
- System is stable iff $|\lambda_i(A)| < 1$

[1] When A non-negative and x_0 non-negative we have Markov chains.

Stability of Linear Systems

SDP viewpoint:

- Lyapunov functions (generalize *energy* in systems). Functions on $x(t)$ decrease monotonically on trajectories of the system.

Stability of Linear Systems

SDP viewpoint:

- Lyapunov functions (generalize *energy* in systems). Functions on $x(t)$ decrease monotonically on trajectories of the system.
- For our discrete-time system, we have:

$$V(x(t)) = x(t)^T P x(t)$$

$$P \succ 0$$

Stability of Linear Systems

SDP viewpoint:

- Lyapunov functions (generalize *energy* in systems). Functions on $x(t)$ decrease monotonically on trajectories of the system.
- For our discrete-time system, we have:

$$V(x(t)) = x(t)^T P x(t)$$

- To make these monotonically decreasing, we need:

$$\begin{aligned} V(x(t+1)) \leq V(x(t)) &\Leftrightarrow x(t+1)^T P x(t+1) - x(t)^T P x(t) \leq 0 \\ &\Leftrightarrow x(t)^T A^T P A x(t) - x(t)^T P x(t) \leq 0 \\ &\Leftrightarrow A^T P A - P \preceq 0 \end{aligned}$$

for any x(t) →

← LMI

Stability of Linear Systems

SDP viewpoint:

- Lyapunov functions (generalize *energy* in systems). Functions on $x(t)$ decrease monotonically on trajectories of the system.
- For our discrete-time system, we have:

$$V(x(t)) = x(t)^T P x(t)$$

- To make these monotonically decreasing, we need:

$$\begin{aligned} V(x(t+1)) \leq V(x(t)) &\Leftrightarrow x(t+1)^T P x(t+1) - x(t)^T P x(t) \leq 0 \\ &\Leftrightarrow x(t)^T A^T P A x(t) - x(t)^T P x(t) \leq 0 \\ &\Leftrightarrow A^T P A - P \preceq 0 \end{aligned}$$

use SDP to find P!

Theorem

Given matrix $A \in \mathbb{R}^{m \times m}$, the following conditions are equivalent:

1. *All eigenvalues of A are inside unit circle, i.e. $|\lambda_i(A)| < 1$*
2. *There is $P \in \mathcal{S}^m$ such that*

$$P \succ 0, \quad A^T P A - P \prec 0$$

LMI on P

Where is the control?

Setup:

- Linear difference equation, with *control input*

$$x(t+1) = Ax(t) + Bu(t), \quad x(0) = x_0$$

where $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{m\times k}$

$u(t) \in R^k$

Where is the control?

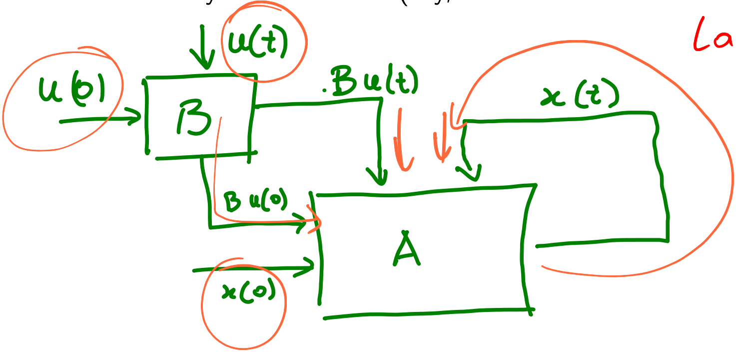
Setup:

- Linear difference equation, with *control input*

$$x(t+1) = Ax(t) + Bu(t), \quad x(0) = x_0$$

  where $A \in \mathbb{R}^{m \times m}$, $B \in \mathbb{R}^{m \times k}$

- If we properly choose control input $u(t)$ we can make our system $x(t)$ behave in a way that we want (say, to stabilize an unstable system)



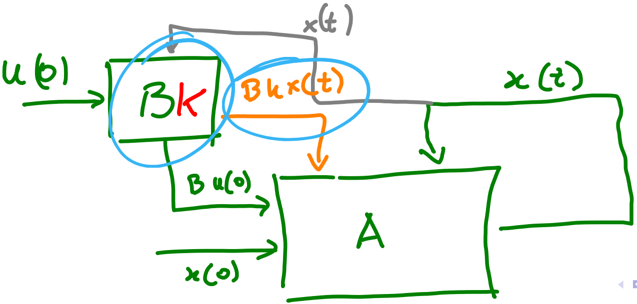
Where is the control?

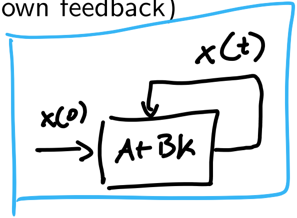
Setup:

- Linear difference equation, with *control input*

$$x(t+1) = Ax(t) + Bu(t), \quad x(0) = x_0$$

  where $A \in \mathbb{R}^{m \times m}$, $B \in \mathbb{R}^{m \times k}$
- If we properly choose control input $u(t)$ we can make our system $x(t)$ behave in a way that we want (say, to stabilize an unstable system)
- Want to do it by setting the control input to be $u(t) = Kx(t)$ for some fixed K (so that we use the system as its own feedback)





Where is the control?

Setup:

- Linear difference equation, with *control input*

$$x(t+1) = Ax(t) + Bu(t), \quad x(0) = x_0$$

  where $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{m\times k}$
- If we properly choose control input $u(t)$ we can make our system $x(t)$ behave in a way that we want (say, to stabilize an unstable system)
- Want to do it by setting the control input to be $u(t) = Kx(t)$ for some fixed K (so that we use the system as its own feedback)
- Same thing as replacing $A \leftarrow A + BK$

Where is the control?

Setup:

- Linear difference equation, with *control input*

$$x(t+1) = Ax(t) + Bu(t), \quad x(0) = x_0$$

  where $A \in \mathbb{R}^{m \times m}$, $B \in \mathbb{R}^{m \times k}$
- If we properly choose control input $u(t)$ we can make our system $x(t)$ behave in a way that we want (say, to stabilize an unstable system)
- Want to do it by setting the control input to be $u(t) = Kx(t)$ for some fixed K (so that we use the system as its own feedback)
- Same thing as replacing $A \leftarrow A + BK$
- Now this is harder to solve via simple eigenvalue description. But still solved the same way via Lyapunov functions!

Where is the control?

Setup:

- Linear difference equation, with *control input*

$$x(t+1) = Ax(t) + Bu(t), \quad x(0) = x_0$$

  where $A \in \mathbb{R}^{m \times m}$, $B \in \mathbb{R}^{m \times k}$
- If we properly choose control input $u(t)$ we can make our system $x(t)$ behave in a way that we want (say, to stabilize an unstable system)
- Want to do it by setting the control input to be $u(t) = Kx(t)$ for some fixed K (so that we use the system as its own feedback)
- Same thing as replacing $A \leftarrow A + BK$
- Now this is harder to solve via simple eigenvalue description. But still solved the same way via Lyapunov functions!
- Want $P \succ 0$ such that

variables

$P \succ 0$

B, k

$$(A+BK)^T P (A+BK) - P \prec 0$$

not LMI!

Where is the control?

Setup:

- Linear difference equation, with *control input*

$$x(t+1) = Ax(t) + Bu(t), \quad x(0) = x_0$$

  where $A \in \mathbb{R}^{m \times m}$, $B \in \mathbb{R}^{m \times k}$
- If we properly choose control input $u(t)$ we can make our system $x(t)$ behave in a way that we want (say, to stabilize an unstable system)
- Want to do it by setting the control input to be $u(t) = Kx(t)$ for some fixed K (so that we use the system as its own feedback)
- Same thing as replacing $A \leftarrow A + BK$
- Now this is harder to solve via simple eigenvalue description. But still solved the same way via Lyapunov functions!
- Want $P \succ 0$ such that

$$(A+BK)^T P(A+BK) - P \prec 0$$

- Wait, this ain't no SDP! But we can make it into SDP with some matrix manipulations.

Conclusion

- Mathematical programming - very general, and pervasive in Algorithmic life

[2] pun intended

Conclusion

- Mathematical programming - very general, and pervasive in Algorithmic life
- General mathematical programming very hard

[2] pun intended

Conclusion

- Mathematical programming - very general, and pervasive in Algorithmic life
- General mathematical programming very hard
- Special cases have very striking applications!
  - Linear Programming (previous lectures)
  - Today: *Semidefinite Programming*

[2] pun intended

Conclusion

- Mathematical programming - very general, and pervasive in Algorithmic life
- General mathematical programming very hard
- Special cases have very striking applications!
  - Linear Programming (previous lectures)
  - Today: *Semidefinite Programming*
- Semidefinite Programming and Duality - fundamental concepts, lots of applications!

[2] pun intended

Conclusion

- Mathematical programming - very general, and pervasive in Algorithmic life
- General mathematical programming very hard
- Special cases have very striking applications!
  - Linear Programming (previous lectures)
  - Today: *Semidefinite Programming*
- Semidefinite Programming and Duality - fundamental concepts, lots of applications!
  - Applications in Combinatorial Optimization (Max-Cut in *next lecture*!)
  - Applications in Control Theory (today)
  - many more!

[2] pun intended

Conclusion

- Mathematical programming - very general, and pervasive in Algorithmic life
- General mathematical programming very hard
- Special cases have very striking applications!
  - Linear Programming (previous lectures)
  - Today: *Semidefinite Programming*
- Semidefinite Programming and Duality - fundamental concepts, lots of applications!
  - Applications in Combinatorial Optimization (Max-Cut in *next lecture*!)
  - Applications in Control Theory (today)
  - many more!
- Check out connections to Sum of Squares and a **bold** attempt[2] to have one algorithm to solve all problems! (i.e., one algorithm to rule them all)

https://windowsontheory.org/2016/08/27/proofs-beliefs-and-algorithms-through-the-lens-of-sum-of-squares/

[2] pun intended

Acknowledgement

- Lecture based largely on:
  - [Blekherman, Parrilo, Thomas 2012, Chapter 2]

Ryan O'Donnell lecture on SDPs

References I

Blekherman, Grigoriy and Parrilo, Pablo and Thomas, Rekha (2012)
Convex Algebraic Geometry