ECE 490: Introduction to Optimization **Spring 2022**

Lecture 27

Semidefinite Programming

Lecturer: Bin Hu, Date:04/29/2022

In this lecture, we give an overview of semidefinite programming (SDP). This is an important class of convex optimization problems. We will talk about the problem formulation, the dual form, and several applications.

27.1 Review: Constrained Optimization

Recall that in general an optimization problem has the following form:

$$\min_{x \in X} f(x)$$

where x is the decision variable, f is the objective function, and X is some feasible set. In most situations, the feasible set X is characterized by some equality and inequality constraints. This leads to the following general formulation for constrained optimization:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \ i = 1, \ldots, m \\ & h_j(x) = 0, \ j = 1, \ldots, l \end{aligned} \tag{27.1}$$

We have inequality constraints in the form of $g_i(x) \leq 0$ and equality constraints in the form of $h_j(x) = 0$. The optimal value of the above problem is defined as $f^* = \inf\{f(x) | g_i(x) \leq 0, \, i = 1, \ldots, m, \, h_j(x) = 0, \, j = 1, \ldots, l\}$. If there does not exist any x satisfying $g_i(x) \leq 0$ and $h_j(x) = 0$ for all (i, j), we have $f^* = \infty$. In this case, we say the problem (27.1) is infeasible. If the problem (27.1) is unbounded below, we have $f^* = -\infty$. For example, consider the following problem

$$\begin{aligned} \text{minimize} \quad & x + y \\ \text{subject to} \quad & x^2 + y^2 \leq 1 \\ & x + y = 100 \end{aligned}$$

We cannot find any real number pair (x, y) to satisfy $x^2 + y^2 \leq 1$ and $x + y = 100$ simultaneously. For this case we have $f^* = \infty$.

Let's look at another example. We consider the following simple problem

$$\begin{aligned} \text{minimize} \quad & x \\ \text{subject to} \quad & x \leq 1 \end{aligned}$$

In this case, the problem is unbounded below. Hence we have $f^* = -\infty$.

Clearly, the forms of g_i and h_j will affect whether we can efficiently solve (27.1) or not. Depending on the properties of g_i and h_j, the problem (27.1) can become very challenging. That is why it is desired to formulate problems as convex optimization.

The so-called feasibility problem can be viewed as a special case of the general constrained optimization problem. Specifically, the feasibility problem has the following form:

$$\begin{aligned} \text{find} \quad & x \\ \text{subject to} \quad & g_i(x) \leq 0,\ i = 1, \ldots, m \\ & h_j(x) = 0,\ j = 1, \ldots, l \end{aligned} \tag{27.2}$$

It asks whether we can find a point x such that the constraints $g_i(x) \leq 0$ and $h_j(x) = 0$ are satisfied for all (i, j). This can be considered as a special case of (27.1) where f is set to be any constant function. For example, (27.2) can be reformulated as

$$\begin{aligned} \text{minimize} \quad & 0 \\ \text{subject to} \quad & g_i(x) \leq 0,\ i = 1, \ldots, m \\ & h_j(x) = 0,\ j = 1, \ldots, l \end{aligned} \tag{27.3}$$

By the above formulation, we have $f^* = 0$ if the original problem (27.2) is feasible, and $f^* = \infty$ if (27.2) is infeasible.

27.2 What is SDP?

For convex optimization, we have linear equality constraints and g_i is convex for all i. Notice any equality constraint $h_j(x) = 0$ can be equivalently rewritten as two inequality constraints $h_j(x) \leq 0$ and $h_j(x) \geq 0$. Therefore, in convex programming, we expect h_j and $-h_j$ are both convex. This naturally leads to the linear equality constraints. Standard convex programming problems include:

- Linear programming (LP): We consider a more general form of LP.

$$\begin{aligned} \text{minimize} \quad & c^\mathsf{T} x \\ \text{subject to} \quad & Gx - r \leq 0 \\ & Ax - b = 0 \end{aligned} \tag{27.4}$$

  where $Gx - r \leq 0$ means all the entries of the vector $(Gx - r)$ are non-positive. Many problems can be reformulated as LP problems. For example, the piecewise-linear minimization problem minimize $max_{i=1,\ldots,m}(a_i^\mathsf{T} x + b_i)$ can be reformulated as

$$\begin{aligned} \text{minimize} \quad & t \\ \text{subject to} \quad & a_i^\mathsf{T} x + b_i \leq t,\ i = 1, \ldots, m \end{aligned} \tag{27.5}$$

which is in the form of (27.4) if we augment x and t as our new decision variable vector and choose (c, G, r) as

$$c = \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix}, \; G = \begin{bmatrix} a_1^\top & -1 \\ a_2^\top & -1 \\ \vdots & -1 \\ a_m^\top & -1 \end{bmatrix}, \; r = \begin{bmatrix} -b_1 \\ -b_2 \\ \vdots \\ -b_m \end{bmatrix}.$$

Many solvers are available for LPs. Both the simplex method and the interior-point methods have been implemented for practical large scale problems. In Matlab, you can use the function linprog to solve LPs.

- Second-order cone programming (SOCP):

$$\begin{aligned} \text{minimize} \quad & c^\top x \\ \text{subject to} \quad & \|F_i x + d_i\| \le f_i^\top x + r_i, \; i = 1, \dots, m \\ & Ax - b = 0 \end{aligned} \tag{27.6}$$

SOCP is more general than LP but also slightly more difficult. One example for SOCP is the so-called robust LP problem. There are also many solvers for SOCP.

- Semidefinite programming (SDP):

$$\begin{aligned} \text{minimize} \quad & c^\top x \\ \text{subject to} \quad & x_1 F_1 + x_2 F_2 + \cdots + x_p F_p + F_0 \preceq 0 \\ & Ax - b = 0 \end{aligned} \tag{27.7}$$

where $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$ is the decision variable vector, F_i are symmetric matrices for all i, and the matrix inequality holds in the semidefinite sense. Hence we want to find x such that $x_1 F_1 + x_2 F_2 + \cdots + x_p F_p + F_0$ is a negative semidefinite matrix. SDP is very powerful. We will see a few examples in the next section.

27.3 Examples

The first example is the eigenvalue minimization problem. Suppose we want to minimize $\lambda_{\max}(A(x))$ where $A(x) = A_0 + x_1 A_1 + x_2 A_2 + \ldots + x_n A_n$. We know $\lambda_{\max}(A) \le t$ if and only if $A \preceq tI$. Therefore, the eigenvalue minimization problem is equivalent to the following SDP

$$\begin{aligned} \text{minimize} \quad & t \\ \text{subject to} \quad & A(x) \preceq tI \end{aligned} \tag{27.8}$$

The above problem is a SDP with the new decision variable $\tilde{x} = \begin{bmatrix} t & x_1 & x_2 & \dots & x_n \end{bmatrix}^\mathsf{T}$. Then (27.8) can be rewritten as

$$\begin{aligned} \text{minimize} \quad & \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \end{bmatrix} \tilde{x} \\ \text{subject to} \quad & A_0 - tI + x_1 A_1 + x_2 A_2 + \dots + x_n A_n \preceq 0. \end{aligned} \tag{27.9}$$

Therefore, the eigenvalue minimization problem can be formulated as a SDP.

Next, we look at the matrix norm minimization problem. Suppose now we want to minimize $\|A(x)\| = \sqrt{\lambda_{\max}(A(x)^\mathsf{T} A(x))}$, where $A(x) = A_0 + x_1 A_1 + x_2 A_2 + \dots + x_n A_n$ with given $A_i \in \mathbb{R}^{p \times q}$. Here $A(x)$ is not necessarily a square matrix. We know $\|A\| \leq t$ if and only if $A^\mathsf{T} A \preceq t^2 I$ and $t \geq 0$. By Schur complement lemma, $A^\mathsf{T} A \preceq t^2 I$ and $t \geq 0$ if and only if

$$\begin{bmatrix} tI & A \\ A^\mathsf{T} & tI \end{bmatrix} \succeq 0.$$

Hence the matrix norm minimization is equivalent to the following SDP

$$\begin{aligned} \text{minimize} \quad & t \\ \text{subject to} \quad & \begin{bmatrix} tI & A(x) \\ A(x)^\mathsf{T} & tI \end{bmatrix} \succeq 0, \end{aligned} \tag{27.10}$$

whose decision variable is $\tilde{x} = \begin{bmatrix} t & x_1 & x_2 & \dots & x_n \end{bmatrix}^\mathsf{T}$. Why is (27.10) a SDP? It can be rewritten as

$$\begin{aligned} \text{minimize} \quad & \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \end{bmatrix} \tilde{x} \\ \text{subject to} \quad & \begin{bmatrix} 0 & -A_0 \\ -A_0^\mathsf{T} & 0 \end{bmatrix} + t \begin{bmatrix} -I & 0 \\ 0 & -I \end{bmatrix} + x_1 \begin{bmatrix} 0 & -A_1 \\ -A_1^\mathsf{T} & 0 \end{bmatrix} + \dots + x_n \begin{bmatrix} 0 & -A_n \\ -A_n & 0 \end{bmatrix} \preceq 0. \end{aligned}$$

Our last example is from control theory. We aim at analyzing the stability of the so-called (autonomous) linear time-invariant system

$$\xi_{k+1} = A \xi_k \tag{27.11}$$

Here we consider discrete-time systems, and ξ_k is the state at time step k. Given the initial condition ξ_0, then the sequence $\{\xi_k\}$ is completely determined by (27.11). One fundamental question control people usually ask is whether (27.11) is stable. The system (27.11) is internally stable if ξ_k converges to 0 given any arbitrary initial condition ξ_0. Notice (27.11) just states that we have $\xi_k = A^k \xi_0$. There are four possible cases.

1. When A is Schur stable (or equivalently the spectral radius of A is smaller than 1), the term A^k converges to a zero matrix. For example, if $A = \begin{bmatrix} 0.9 & 0 \\ 0 & 0.9 \end{bmatrix}$, then $A^k = 0.9^k I$.

2. When the spectral radius of A is equal to 1 and all the Jordan blocks corresponding to eigenvalues with magnitude equal to 1 are 1×1, A^k remains bounded for any k. This is the so-called marginal stability case where $A^k\xi_0$ remains bounded but may not converge to 0. For example, if $A = \begin{bmatrix} 1 & 0 \\ 0 & 0.9 \end{bmatrix}$, then $A^k = \begin{bmatrix} 1 & 0 \\ 0 & 0.9^k \end{bmatrix}$.

3. When the spectral radius of A is equal to 1 and at least one of the Jordan blocks corresponding to eigenvalues with magnitude equal to 1 is not 1×1, A^k becomes unbounded. For example, if $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, then $A^k = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}$.

4. When the spectral radius of A is larger than 1, A^k also becomes unbounded. For example, if $A = \begin{bmatrix} 1.1 & 0 \\ 0 & 1.1 \end{bmatrix}$, then $A^k = 1.1^k I$.

Therefore, it is straightforward to verify that (27.11) is stable if and only if the spectral radius of A is strictly less than 1. The spectral radius condition only works for such linear time-invariant (LTI)[1] system. It is hard to extend such conditions for time-varying or nonlinear systems. Alternatively, one can also formulate necessary and sufficient stability conditions for (27.11) using semidefinite programs.

Theorem 27.1. *The system* (27.11) *is internally stable if and only if there exists a positive definite matrix P such that*

$$A^\mathsf{T} P A - P \prec 0 \tag{27.12}$$

Here the inequality holds in the definite sense (so what we really mean here is that the matrix $(A^\mathsf{T} P A - P)$ needs to be a negative definite matrix).

Proof: We will only show sufficiency since this direction can be generalized for time-varying or nonlinear systems. If (27.12) holds, then there exists a sufficiently small positive number $\varepsilon > 0$ such that $A^\mathsf{T} P A - P \preceq -\varepsilon P$ which can be rewritten as

$$A^\mathsf{T} P A - (1-\varepsilon) P \preceq 0.$$

Therefore we can left and right multiply both sides of the above inequality with ξ_k^T and ξ_k and obtain

$$(A\xi_k)^\mathsf{T} P (A\xi_k) - (1-\varepsilon)\xi_k^\mathsf{T} P \xi_k \leq 0$$

We have $\xi_{k+1} = A\xi_k$ and the above inequality is equivalent to $\xi_{k+1}^\mathsf{T} P \xi_{k+1} \leq (1-\varepsilon)\xi_k^\mathsf{T} P \xi_k$. By induction, we have

$$\xi_k^\mathsf{T} P \xi_k \leq (1-\varepsilon)^k \xi_0^\mathsf{T} P \xi_0$$

[1]This just means A is a constant matrix and does not change over time.

Since P is positive definite, we have $\xi_k^\mathsf{T} P \xi_k \geq \lambda_{\min}(P)\|\xi_k\|^2$ where $\lambda_{\min}(P)$ is the smallest eigenvalue of P and is a positive number. Finally we have

$$\|\xi_k\|^2 \leq (1-\varepsilon)^k c \tag{27.13}$$

where $c = \frac{\xi_0^\mathsf{T} P \xi_0}{\lambda_{\min}(P)}$. We know $0 \leq 1-\varepsilon < 1$ and hence $\|\xi\|$ converges to 0 as k goes to ∞. We have established the internal stability of (27.11).

The proof for necessity relies on the LTI assumption and is omitted here. ■

How to use the condition (27.12)**?** The testing condition (27.12) leads to a semidefinite program. Given A, the left side of (27.12) is linear in P. One just needs to search such positive definite P for this feasibility problem.

27.4 Dual of SDP

Now we consider the following semidefinite program (SDP) problem.

$$\begin{aligned} \text{minimize} \quad & c^\mathsf{T} x \\ \text{subject to} \quad & x_1 F_1 + x_2 F_2 + \ldots + x_p F_p - G \preceq 0. \end{aligned} \tag{27.14}$$

Here $x \in \mathbb{R}^p$ and we have

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$$

where x_i $(i = 1, 2, \ldots, p)$ is just scaler. Here F_i $(i = 1, 2, \ldots, p)$ and G are all symmetric matrices. The inequality $x_1F_1 + x_2F_2 + \ldots + x_pF_p - G \preceq 0$ just means $(x_1F_1 + x_2F_2 + \ldots + x_pF_p - G)$ is a negative semidefinite matrix. To derive the dual problem for SDP, we need the matrix version of Lagrangian formulations. Recall that the term $\mu^\mathsf{T} g(x)$ in the Lagrangian can be viewed as an inner product between the Lagrangian multiplier μ and the constraint function $g(x)$. For the SDP problem, the Lagrangian multiplier is a matrix Y and the inner product between Y and $(x_1F_1 + x_2F_2 + \ldots + x_pF_p - G)$ is $\text{trace}(Y(x_1F_1 + x_2F_2 + \ldots + x_pF_p - G))$. Let's explain the inner product of two matrices first. Consider two symmetric matrices A and B. If we put augment the entries of A as a vector and also augment all the entries of B as a vector, then clearly the inner product of these two resultant vectors is $\sum_{i,j} A_{ij}B_{ij}$. This sum can be compactly rewritten as $\text{trace}\, AB$ where trace just denotes the sum of the diagonal entries of a given matrix. Therefore, the Lagrangian for (27.14) can be written as

$$\begin{aligned} L(x, Y) =& c^\mathsf{T} x + \text{trace}(Y(x_1F_1 + x_2F_2 + \ldots + x_pF_p - G)) \\ =& -\text{trace}(YG) + \sum_{i=1}^p x_i(c_i + \text{trace}(YF_i)) \end{aligned}$$

where c_i is the i-th entry of c. We have

$$D(Y) = \min_{x \in \mathbb{R}^p} L(x, Y) = \begin{cases} -\operatorname{trace}(YG) & \text{if } c_i + \operatorname{trace}(YF_i) = 0 \\ -\infty & \text{Otherwise} \end{cases} \tag{27.15}$$

Therefore, the dual problem for SDP is

$$\begin{aligned} \text{maximize} \quad & -\operatorname{trace}(GY) \\ \text{subject to} \quad & \operatorname{trace}(F_iY) + c_i = 0, \ \forall i = 1, \ldots, p \\ & Y \succeq 0 \end{aligned}$$

Here $Y \succeq 0$ just states that Y is a positive semidefinite matrix. Clearly G, F_i, and c_i are all given, and Y is the decision variable for this dual problem.

27.5 Solvers for SDP

When implementing the interior point method, the log det barrier function is typically used. However, typically we don't need to implement the interior point method by ourselves. We can use `cvx` (in Matlab) or `cvxpy` (for Python).