

Lecture 8 Conic Programming

Can Li

ChE 597: Computational Optimization
Purdue University

Why conic programming?

- Conic programming including Linear, Conic Quadratic and Semidefinite Programming can be seen as a special class of convex optimization problems that are “well-structured”.
- Wide range of applications in fields like control theory, finance, signal processing, engineering, and machine learning

Cone, convex cone, conic combination

Cone: $C \subseteq \mathbb{R}^n$ such that

$$x \in C \implies tx \in C \text{ for all } t \geq 0$$



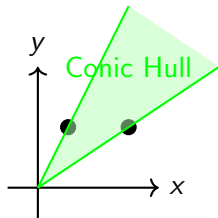
Convex cone: cone that is also convex, i.e.,

$$x_1, x_2 \in C \implies t_1x_1 + t_2x_2 \in C \text{ for all } t_1, t_2 \geq 0$$

Conic combination of $x_1, \dots, x_k \in \mathbb{R}^n$:

$$\lambda_1x_1 + \dots + \lambda_kx_k$$

with $\lambda_i \geq 0, i = 1, \dots, k$.

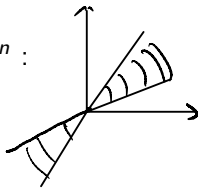


Conic hull collects all conic combinations

Proper convex cones

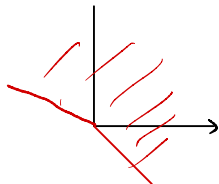
We consider proper convex cones \mathcal{K} in \mathbb{R}^n :

- Closed
- Pointed: $\mathcal{K} \cap (-\mathcal{K}) = \{0\}$.
- Non-empty interior.



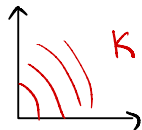
Dual-cone:

$$\mathcal{K}^* = \left\{ v \in \mathbb{R}^n \mid u^T v \geq 0, \forall u \in \mathcal{K} \right\}$$



If \mathcal{K} is a proper cone, then \mathcal{K}^* is also proper.

We use the notation:



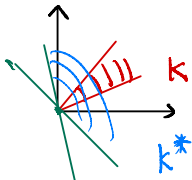
$\mathcal{K}^* = \mathcal{K}$
self-dual

$$x \succeq_{\mathcal{K}} y \iff (x - y) \in \mathcal{K}$$

$$x \succ_{\mathcal{K}} y \iff (x - y) \in \text{int } \mathcal{K}$$

if \mathcal{K} is first
or third.

$$x \succeq_{\mathcal{K}} y \iff x \succeq y$$



What is conic programming?

Let \mathcal{K} be a proper convex cone in \mathbb{R}^m (pointed, closed, and with a nonempty interior). Given an objective $c \in \mathbb{R}^n$, an $m \times n$ constraint matrix A , and a right-hand side $b \in \mathbb{R}^m$, consider the optimization problem

$$\min_x \left\{ c^T x \mid Ax \succeq_{\mathcal{K}} b \right\}$$

where $Ax \succeq_{\mathcal{K}} b \Leftrightarrow Ax - b \in \mathcal{K}$

- Linear program is a special case of conic program (why?).
For LP, $\mathcal{K} = \{y \in \mathbb{R}^m \mid y \geq 0\}$.

Dual of conic program

The dual of the conic program is

$$\max \left\{ b^T \lambda \mid A^T \lambda = c, \lambda \succeq_{\mathcal{K}^*} 0 \right\}$$

where \mathcal{K}^* represents the dual cone of \mathcal{K} .

- Note that for LP, \mathcal{K} is self-dual. $\mathcal{K} = \mathcal{K}^* = \{y \in \mathbb{R}^m \mid y \geq 0\}$. We reproduce LP duality!
- The duality is **symmetric**: the dual problem is conic, and the problem dual to dual is (equivalent to) the primal.
- **Weak duality**: value of the dual objective at every dual feasible solution λ is the value of the primal objective at every primal feasible solution x , so that the duality gap

$$c^T x - b^T \lambda \geq 0$$

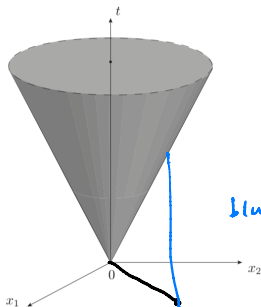
at every primal-dual feasible pair (x, λ) .

Strong duality

- If the primal is bounded below and strictly feasible, i.e., $Ax \succ_{\mathcal{K}} b$ for some x , then the dual is solvable and the optimal values in the problems are equal to each other.
- If the dual is bounded above and strictly feasible (i.e., exists $\lambda \succ_{\mathcal{K}^*} 0$ such that $A^T \lambda = c$), then the primal is solvable and we have strong duality.
- Assume that at least one of the problems is bounded and strictly feasible. Then a primal-dual feasible pair (x, λ) is a pair of optimal solutions to the respective problems if and only if the following condition hold
 - $b^T \lambda = c^T x$ [zero duality gap]
 - $\lambda^T [Ax - b] = 0$ [complementary slackness].

Second-order cone

- The second-order cone (SOC) in \mathbb{R}^3 is the set of vectors (x_1, x_2, t) such that $\sqrt{x_1^2 + x_2^2} \leq t$. Horizontal sections of this set at level $\alpha \geq 0$ are disks of radius α .



$$\forall x_1, x_2, t.$$

$$\sqrt{x_1^2 + x_2^2} \leq t.$$

$$(\lambda x_1, \lambda x_2, \lambda t) \in K.$$

$$\sqrt{(\lambda x_1)^2 + (\lambda x_2)^2}$$

$$\text{blue} = \text{black.} \quad = \lambda \sqrt{x_1^2 + x_2^2} \leq \lambda t.$$

- In arbitrary dimension: an $(n+1)$ -dimensional SOC is the following set:

$$\mathcal{K}_n = \{(x, t), x \in \mathbb{R}^n, t \in \mathbb{R} : \|x\|_2 \leq t\}.$$

- SOC is self-dual: $\mathcal{K}_n^* = \mathcal{K}_n$

Rotated second order cone

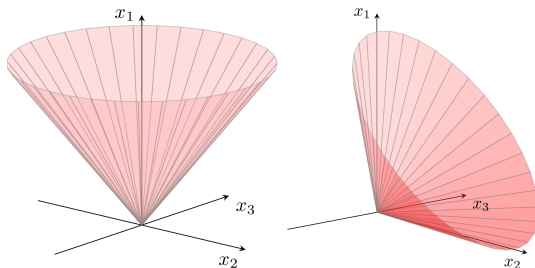


Figure: Boundary of second order cone $x_1 \geq \sqrt{x_2^2 + x_3^2}$ and rotated second cone $2x_1x_2 \geq x_3^2, x_1, x_2 \geq 0$.

Rotated second order cone

- The rotated second-order cone in \mathbb{R}^{n+2} is the set

$$\mathcal{K}_n^r = \left\{ (x, y, z), x \in \mathbb{R}^n, y \in \mathbb{R}, z \in \mathbb{R} : x^\top x \leq 2yz, y \geq 0, z \geq 0 \right\}.$$

- The rotated second-order cone in \mathbb{R}^{n+2} can be expressed as a linear transformation (actually, a rotation) of the (plain) second-order cone in \mathbb{R}^{n+2} , since

$$\|x\|_2^2 \leq 2yz, y \geq 0, z \geq 0 \iff \left\| \begin{bmatrix} x \\ \frac{1}{\sqrt{2}}(y - z) \end{bmatrix} \right\|_2 \leq \frac{1}{\sqrt{2}}(y + z).$$

That is, $(x, y, z) \in \mathcal{K}_n^r$ if and only if $(w, t) \in \mathcal{K}_n$, where

$$w = (x, (y - z)/\sqrt{2}), \quad t = (y + z)/\sqrt{2}.$$

$$\left\| \frac{x}{\sqrt{2}} (y-z) \right\|_2 \leq \frac{1}{\sqrt{2}} (y+z)$$

$$y+z \geq 0.$$

$$\left(\frac{x}{\sqrt{2}} (y-z) \right)^T \left(\frac{x}{\sqrt{2}} (y-z) \right) \leq \frac{1}{2} (y+z)^2.$$

$$x^T x + \frac{1}{2} (y-z)^2 \leq \frac{1}{2} (y+z)^2.$$

$$x^T x \leq 2yz.$$

Standard SOC constraint

- The standard format of a second-order cone constraint on a variable $x \in \mathbb{R}^n$ expresses the condition that $(y, t) \in \mathcal{K}_m$, with $y \in \mathbb{R}^m, t \in \mathbb{R}$, where y, t are some affine functions of x .
- These affine functions can be expressed as $y = Ax + b, t = c^\top x + d$, hence the condition $(y, t) \in \mathcal{K}_m$ becomes

$$\|Ax + b\|_2 \leq c^\top x + d,$$

where $A \in \mathbb{R}^{m,n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$, and $d \in \mathbb{R}$.

- For example, the quadratic constraint

$$x^\top Qx + c^\top x \leq t, \quad Q \succeq 0 \quad Q = VV^\top$$

can be expressed in conic form as

$$\left\| \begin{bmatrix} \sqrt{2}Q^{1/2}x \\ t - c^\top x - 1/2 \end{bmatrix} \right\|_2 \leq t - c^\top x + 1/2.$$

Second-order cone programs

- A second-order cone program is a convex optimization problem having linear objective and SOC constraints. When the SOC constraints have the standard form, we have a SOCP in standard inequality form:

$$\begin{array}{ll}\min_{x \in \mathbb{R}^n} & c^\top x \\ \text{s.t.:} & \|A_i x + b_i\|_2 \leq c_i^\top x + d_i, \quad i = 1, \dots, m,\end{array}$$

where $A_i \in \mathbb{R}^{m_i, n}$ are given matrices, $b_i \in \mathbb{R}^{m_i}$, $c_i \in \mathbb{R}^n$ are vectors, and d_i are given scalars.

- SOCPs are representative of a quite large class of convex optimization problems. Indeed, LPs, convex QPs, and convex QCQPs can all be represented as SOCPs.

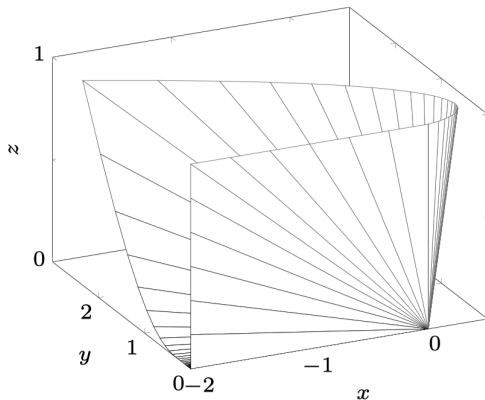
Exponential cone

e^x perspective $\Rightarrow ye^{x/y}$

the epigraph of the perspective of $\exp x$ is a non-proper cone

$$\mathcal{K} = \left\{ (x, y, z) \in \mathbb{R}^3 \mid ye^{x/y} \leq z, y > 0 \right\}$$

the exponential cone is $\mathcal{K}_{\text{exp}} = \text{cl } \mathcal{K} = \mathcal{K} \cup \{(x, 0, z) \mid x \leq 0, z \geq 0\}$



Exponential cone representable functions

- The epigraph $t \geq e^x$ is a section of \mathcal{K}_{exp} :

$$t \geq e^x \iff (x, 1, t) \in \mathcal{K}_{\text{exp}} .$$

- Similarly, we can express the hypograph $t \leq \log x, x \geq 0$:

$$t \leq \log x \iff (t, 1, x) \in \mathcal{K}_{\text{exp}} .$$

- The entropy function $H(x) = -x \log x$ can be maximized using the following representation.:

$$t \leq -x \log x \iff t \leq x \log(1/x) \iff (t, x, 1) \in \mathcal{K}_{\text{exp}} .$$

$$\mathcal{K}_{\text{exp}} \{ (x, y, z) : y \exp(-\frac{x}{y}) \leq z .$$

$$\begin{aligned} x \exp(-\frac{t}{x}) &\leq 1 \\ \log x + \frac{t}{x} &\leq 0 . \end{aligned}$$

Power cone

Definition: for $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m) > 0$ and $\sum_{i=1}^m \alpha_i = 1$

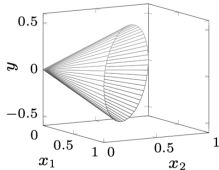
$$K_\alpha = \{(x, y) \in \mathbb{R}_+^m \times \mathbb{R} \mid |y| \leq x_1^{\alpha_1} \cdots x_m^{\alpha_m}\}$$

$$(x, y) \in K_\alpha$$

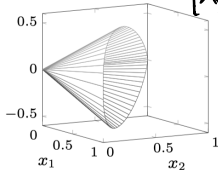
Examples for $m = 2$

$$(\lambda x, \lambda y) \in K_\alpha \quad \forall \lambda \geq 0$$

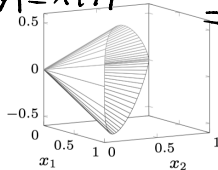
$$\alpha = (\frac{1}{2}, \frac{1}{2})$$



$$\alpha = (\frac{2}{3}, \frac{1}{3})$$



$$| \lambda y | = \lambda | y | \quad \alpha = (\frac{3}{4}, \frac{1}{4})$$



$$(\lambda x_1)^{\alpha_1} \cdots (\lambda x_m)^{\alpha_m} = \lambda^{\alpha_1 + \dots + \alpha_m} = \lambda$$

Positive semidefinite cone

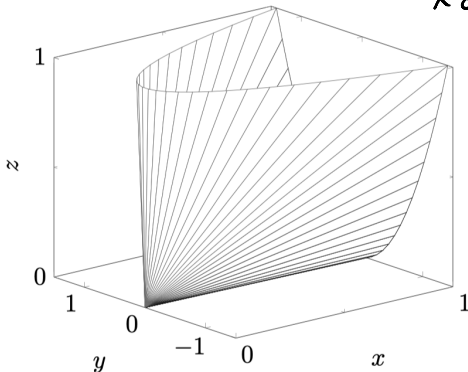
$$\mathcal{S}^p = \{\text{vec}(X) \mid X \in \mathcal{S}_+^p\} = \left\{x \in \mathbb{R}^{p(p+1)/2} \mid \text{mat}(x) \succeq 0\right\}$$

$$\mathcal{S}^2 = \left\{(x, y, z) \mid \begin{bmatrix} x & y/\sqrt{2} \\ y/\sqrt{2} & z \end{bmatrix} \succeq 0\right\}$$

$$x \geq 0, \quad z \geq 0$$

$$xz - \frac{y^2}{2} \geq 0$$

$$y^2 \leq 2xz$$



Basic properties of a PSD matrix

- We denote $n \times n$ symmetric matrices by \mathcal{S}^n .
- Standard inner product for matrices:

$$\langle V, W \rangle := \text{tr}(V^T W) = \sum_{ij} V_{ij} W_{ij} = \text{vec}(V)^T \text{vec}(W)$$

- X is positive semidefinite if and only if
 1. $z^T X z \geq 0, \forall z \in \mathbb{R}^n$.
 2. All the eigenvalues of X are nonnegative.
 3. There exist $V \in \mathbb{R}^{r \times n}$ $r \leq n$ such that $X = V^T V$
- The positive (semi)definite matrices form a cone $(\mathcal{S}_+) \mathcal{S}_{++}$.

Eigenvalue decomposition

The eigenvalue decomposition of a positive semi-definite (PSD) matrix A is given by:

$$A = Q\Lambda Q^T$$

where,

- A is the PSD matrix.
- Q is an orthogonal matrix whose columns are the eigenvectors of A .
- Λ is a diagonal matrix with the eigenvalues of A on its diagonal.
- Q^T is the transpose of Q , ensuring the reconstruction of A through the product $Q\Lambda Q^T$.

Pseudo-Inverse of a PSD Matrix

The pseudo-inverse of a PSD matrix A , denoted A^\dagger , holds special interest due to A 's properties:

$$A^\dagger = Q\Lambda^\dagger Q^T$$

where,

$$A^\dagger A = Q\Lambda^\dagger Q^T Q\Lambda Q^T$$

- $A = Q\Lambda Q^T$ is the eigenvalue decomposition of A , with A being PSD.
 $= Q\Lambda^\dagger \Lambda Q^T = QI_r Q^T$
- Λ^\dagger contains the reciprocals of the non-zero eigenvalues of A on its diagonal and zeros elsewhere. $= I_r$
- Vector product form: Suppose we have an eigenvalue decomposition of $Q = \sum_{i=1}^k \lambda_i q_i q_i^T$ where the λ_i , $i = 1, \dots, k$ are the nonzero eigenvalues. q_i , $i = 1, \dots, k$ are the corresponding orthonormal eigenvectors.

$$Q^\dagger := \sum_{i=1}^k \lambda_i^{-1} q_i q_i^T$$

Schur's Lemma

Given:

- $Q \in \mathcal{S}_p, P \in \mathcal{S}_n$
- $S = [s_1, \dots, s_p] \in \mathbb{R}^{n \times p}$

Lemma Statement: The following are equivalent:

- (i) The matrix $\begin{bmatrix} P & S^\top \\ S & Q \end{bmatrix}$ is positive [resp. semi-]definite.
- (ii) The matrices Q and $P - S^\top Q^\dagger S$ are both positive [resp. semi-]definite, and $s_i \in \mathcal{R}(Q)$ for $i = 1, \dots, p$; these range conditions are automatically satisfied when $Q \succ 0$ (then $\mathcal{R}(Q) = \mathbb{R}^p$).

Implications:

- Establishes a condition under which a block matrix is positive (semi-)definite.
- Highlights the relationship between the definiteness of a block matrix and its submatrices.

SOC as SDP

The SOC constraint can be converted to the PSD constraint,
Consider the SOC constraint:

$$\|x\|_2 \leq t$$

This can be represented as a PSD constraint using the Schur's lemma:

$$\begin{bmatrix} t & x^T \\ x & tI \end{bmatrix} \succeq 0$$

This indicates SDP is more “expressive” than SOCP.

$$t \geq 0$$

$$t - x^T (tI)^{-1} x \geq 0$$

$$t - \frac{1}{t} x^T x \geq 0$$

$$x^T x \leq t^2$$

Dual cone of the PSD cone

Dual cone:

$$(\mathcal{S}_+^n)^* = \{Z \in \mathbb{R}^{n \times n} \mid \langle X, Z \rangle \geq 0, \forall X \in \mathcal{S}_+^n\}.$$

The semidefinite is self-dual: $(\mathcal{S}_+^n)^* = \mathcal{S}_+^n$.

Dual cone of the PSD cone

Dual cone:

$$(\mathcal{S}_+^n)^* = \{Z \in \mathbb{R}^{n \times n} \mid \langle X, Z \rangle \geq 0, \forall X \in \mathcal{S}_+^n\}.$$

The semidefinite is self-dual: $(\mathcal{S}_+^n)^* = \mathcal{S}_+^n$.

Proof: Assume $Z \succeq 0$ so that $Z = U^T U$ and $X = V^T V$.

$$\langle X, Z \rangle = \langle V^T V, U^T U \rangle = \text{tr}(V^T V U^T U) = \text{tr}(U V^T V U^T) = \|U V^T\|_F^2 \geq 0.$$

The equalities come from the following properties of trace

$$\langle A, B \rangle = \text{tr}(A^T B)$$

$$\text{tr}(ABCD) = \text{tr}(BCDA) = \text{tr}(CDAB) = \text{tr}(DABC) \quad (\text{cyclic property})$$

$$\text{tr}(A^T A) = \langle A, A \rangle = \sum_{i,j} a_{i,j}^2 = \|A\|_F^2$$

Conversely assume $Z \not\succeq 0$. Then $\exists w \in \mathbb{R}^n$ such that

$$w^T Z w = \langle w w^T, Z \rangle = \langle X, Z \rangle < 0.$$

Applications of SDP-Eigenvalue optimization

$$F(x) = F_0 + x_1 F_1 + \cdots + x_m F_m, \quad F_i \in \mathcal{S}_m.$$

Linear matrix inequalities

- Minimize largest eigenvalue $\lambda_1(F(x))$:

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & \gamma I \succeq F(x), \end{array}$$

- Maximize smallest eigenvalue $\lambda_n(F(x))$:

$$\begin{array}{ll} \text{maximize} & \gamma \\ \text{subject to} & F(x) \succeq \gamma I, \end{array}$$

- Minimize eigenvalue spread $\lambda_1(F(x)) - \lambda_n(F(x))$:

$$\begin{array}{ll} \text{minimize} & \gamma - \lambda \\ \text{subject to} & \gamma I \succeq F(x) \succeq \lambda I \end{array}$$

Relaxation of QUBO

Consider an quadratic unconstrained binary optimization (QUBO) problem

$$\begin{array}{ll}\text{minimize} & x^T Q x + c^T x \\ \text{subject to} & x_i \in \{0, 1\}, \quad i = 1, \dots, n\end{array}$$

where $Q \in \mathcal{S}^n$ can be indefinite. QUBO is NP-hard in general.

- Rewrite binary constraints $x_i \in \{0, 1\}$ and introduce lifted space variables $X \in \mathcal{S}^n$:

$$X_{ij} = x_i x_j, \quad x_i^2 = x_i \iff X = x x^T, \quad \text{diag}(X) = x.$$

- Semidefinite relaxation:

$$X \succeq x x^T, \quad \text{diag}(X) = x.$$

SDP relaxation of QUBO

Lifted non-convex problem:

$$\begin{array}{ll}\text{minimize} & \langle Q, X \rangle + c^T x \\ \text{subject to} & \text{diag}(X) = x \\ & \underline{X = xx^T.} \\ & X \preceq xx^T\end{array}$$

Note that

$$\begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix} \succeq 0 \Leftrightarrow X \succeq xx^T \text{ (Schur's lemma)}$$

Semidefinite relaxation: $I \succeq 0, \quad X - xx^T \preceq 0.$

$$\begin{array}{ll}\text{minimize} & \langle Q, X \rangle + c^T x \\ \text{subject to} & \text{diag}(X) = x \\ & \begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix} \succeq 0\end{array}$$

Goemans
Williamson

Relaxation is exact if $X = xx^T$.

A simple cutting plane algorithm for solving SDP

Semidefinite programming (SDP) problems can be formulated as:

$$\begin{aligned} \min_X & \langle C, X \rangle \\ \text{s.t. } & \langle A^i, X \rangle \geq b_i, \quad i = 1, \dots, m, \\ & X \succeq 0, \end{aligned}$$

where X is the semidefinite matrix variable, $C \in \mathcal{S}^n$ and $A^i \in \mathcal{S}^n$ are given matrices.

1. Start with only the m linear constraints on X .
2. Compute the eigenvalues and eigenvectors of the current solution \hat{X} . If all eigenvalues are non-negative, \hat{X} is feasible and thus optimal.
3. For each negative eigenvalue λ_i , $i = 1, \dots, r$, use its corresponding eigenvector v^i , $i = 1, \dots, r$ to add a linear cut $(v^i)^T X v^i \geq 0$ that excludes the current infeasible solution \hat{X} .
4. Add the linear constraints and repeat from step 2 until a feasible X is found or the algorithm converges to a solution within a desired tolerance.

Iteration 0.

$$\min \langle C, X \rangle \quad (LP)$$

$$\langle A^i, X \rangle \geq b_i \quad \forall i = 1 \dots m.$$

\hat{X} to be the optimal solution.

①. $\hat{X} \preceq 0 \Rightarrow$ terminate.

②. $\hat{X} \not\preceq 0$, $\lambda_1, \dots, \lambda_r < 0$ be eigenvalues
of \hat{X} .
 $v^1 \dots v^r$ ----- eigenvectors.

$$v_i^T \hat{X} v_i < 0.$$

Add constraints

$$(v_i)^T X v_i \geq 0, \quad \forall i = 1 \dots r.$$

to the LP. then resolve.

note: we are accumulating the cuts for all iterations

Software for conic programming

- Gurobi, Cplex ((MI)SOCP)
- Mosek: mostly commonly used conic solver (SOCP, EXP, POW, SDP).
- COPT (SOCP, SDP)
- other solvers: see
<https://jump.dev/JuMP.jl/stable/installation/>
<https://plato.asu.edu/bench.html>
- most of the algorithms for conic programs are variants of the interior point algorithm.
- Although SDP is convex, solving it is usually “slow”.
 $n = 100 - 1,000$ are the problem size we are usually talking about.

References

- Ben-Tal, A., & Nemirovski, A. (2001). Lectures on modern convex optimization: analysis, algorithms, and engineering applications. Society for industrial and applied mathematics.
- Mosek tutorials <https://docs.mosek.com/latest/pythonapi/optimization-tutorials.html>
- L. Vandenbergh, conic optimization <https://www.seas.ucla.edu/~vandenbe/236C/lectures/conic.pdf>
- Lemaréchal, C., & Oustry, F. (1999). Semidefinite relaxations and Lagrangian duality with application to combinatorial optimization (Doctoral dissertation, INRIA).
- L. Ghaoui, Second order cone models https://inst.eecs.berkeley.edu/~ee127/fa19/Lectures/12_socp.pdf