EECS 127/227AT Optimization Models in Engineering
Spring 2020 Discussion 11

1. Dual norms and SOCP

Consider the problem

$$p^* = \min_{\vec{x} \in \mathbb{R}^n} \|A\vec{x} - \vec{y}\|_1 + \mu \|\vec{x}\|_2,$$

where $A \in \mathbb{R}^{m \times n}$, $\vec{y} \in \mathbb{R}^m$, and $\mu > 0$.

(a) Express this (primal) problem in standard SOCP form.

Solution: Introducing slack variables $\vec{z} \in \mathbb{R}^m$, $t \in \mathbb{R}$, we can write

$$\begin{aligned} \min_{\vec{x}, \vec{z}, t} \ & \vec{z}^\top \vec{1} + \mu t \\ \text{s.t.} \ & |(A\vec{x} - \vec{y})_i| \le z_i, \quad i = 1, \ldots, m \\ & \|\vec{x}\|_2 \le t. \end{aligned}$$

This expression now satisfies our definition of an SOCP: the objective is linear, and all constraints are SOC constraints. (To see this, recall that the first set of absolute value constraints can be written equivalently as $(A\vec{x} - \vec{y})_i \le z_i$ and $(A\vec{x} - \vec{y})_i \ge -z_i$, which are linear constraints; we keep the absolute value version for conciseness. The second set of constraints are SOC constraints in canonical form.)

(b) Find a dual to the problem and express it in standard SOCP form.

Hint: Recall that for every vector $\vec{z}$, the following dual norm equalities hold:

$$\|\vec{z}\|_2 = \max_{\vec{u} \,:\, \|\vec{u}\|_2 \le 1} \vec{u}^\top \vec{z}, \quad \|\vec{z}\|_1 = \max_{\vec{u} \,:\, \|\vec{u}\|_\infty \le 1} \vec{u}^\top \vec{z}.$$

Solution: Using the hint, we can rewrite the objective function of the original problem as

$$\|A\vec{x} - \vec{y}\|_1 + \mu \|\vec{x}\|_2 = \max_{\vec{u}: \|\vec{u}\|_\infty \le 1} \vec{u}^\top (A\vec{x} - \vec{y}) + \mu \max_{\vec{v}: \|\vec{v}\|_2 \le 1} \vec{v}^\top \vec{x}.$$

We can then express the original (primal) problem as

$$p^* = \min_{\vec{x}} \max_{\vec{u}, \vec{v}: \|\vec{u}\|_\infty \le 1, \|\vec{v}\|_2 \le 1} \vec{u}^\top (A\vec{x} - \vec{y}) + \mu \vec{v}^\top \vec{x}.$$

To form the dual, we reverse the order of min and max:

$$\begin{aligned} d^* &= \max_{\vec{u}, \vec{v}: \|\vec{u}\|_\infty \le 1, \|\vec{v}\|_2 \le 1} \min_{\vec{x}} \vec{u}^\top (A\vec{x} - \vec{y}) + \mu \vec{v}^\top \vec{x} \\ &\doteq \max_{\vec{u}, \vec{v}: \|\vec{u}\|_\infty \le 1, \|\vec{v}\|_2 \le 1} g(\vec{u}, \vec{v}), \end{aligned}$$

where g is defined as

$$\begin{aligned} g(\vec{u},\vec{v}) &\doteq \min_{\vec{x}} \ \vec{u}^\top(A\vec{x}-\vec{y}) + \mu\vec{v}^\top\vec{x} \\ &= \min_{\vec{x}} \ (\vec{u}^\top A + \mu\vec{v}^\top)\vec{x} - \vec{u}^\top\vec{y} \\ &= \begin{cases} -\vec{u}^\top\vec{y} & \text{if } A^\top\vec{u} + \mu\vec{v} = \vec{0} \\ -\infty & \text{otherwise.} \end{cases} \end{aligned}$$

We can thus rewrite the dual problem as

$$\begin{aligned} d^* = \max_{\vec{u},\vec{v}} \quad & -\vec{u}^\top\vec{y} \\ \text{s.t.} \quad & A^\top\vec{u} + \mu\vec{v} = \vec{0} \\ & \|\vec{u}\|_\infty \leq 1, \ \ \|\vec{v}\|_2 \leq 1. \end{aligned}$$

Noting that the first constraint fully restricts the value of $\vec{v}$ — rewriting it, $\vec{v} = -\frac{A^\top\vec{u}}{\mu}$ — we can plug this value into the third constraint and eliminate $\vec{v}$ from our optimization altogether:

$$\begin{aligned} d^* = \max_{\vec{u}} \quad & -\vec{u}^\top\vec{y} \\ \text{s.t.} \quad & \|A^\top\vec{u}\|_2 \leq \mu \\ & \|\vec{u}\|_\infty \leq 1, \end{aligned}$$

generating our final SOCP dual. If desired, we can further rewrite the final constraint as $\|\vec{u}\|_\infty = \max_i |u_i| \leq 1 \Leftrightarrow |u_i| \leq 1, \ i = 1,\ldots,m \Leftrightarrow u_i \leq 1$ and $u_i \geq -1, \ i = 1,\ldots,m$ to make the linearity of that constraint more explicit.

(c) Assume strong duality holds[1] and that $m = 100$ and $n = 10^6$, i.e., A is 100×10^6. Which problem would you choose to solve using a numerical solver: the primal or the dual? Justify your answer.

Solution: To determine the rough computational complexity of each problem, we examine the number of variables and the number of constraints in each problem. The primal SOCP has $\sim 10^6$ variables and 201 constraints, while the dual has 100 variables and 201 constraints. The dual problem is thus much more efficient to solve.

2. Squaring SOCP constraints

When considering a second-order cone (SOC) constraint, you might be tempted to square it to obtain a classical convex quadratic constraint. This problem explores why that might not always work, and how to introduce additional constraints to maintain equivalence and convexity.

(a) For $\vec{x} \in \mathbb{R}^2$, consider the constraint

$$x_1 - 2x_2 \geq \|\vec{x}\|_2,$$

and its squared counterpart

$$(x_1 - 2x_2)^2 \geq \|\vec{x}\|_2^2.$$

Are the two sets equivalent? Are they both convex?

[1] In fact, you can show that strong duality holds using Sion's theorem, a generalization of the minimax theorem that is beyond the scope of this class.

Solution: The set defined by the first constraint is an SOC constraint and is thus convex. The second, squared constraint

$$(x_1 - 2x_2)^2 \geq \|\vec{x}\|_2^2$$

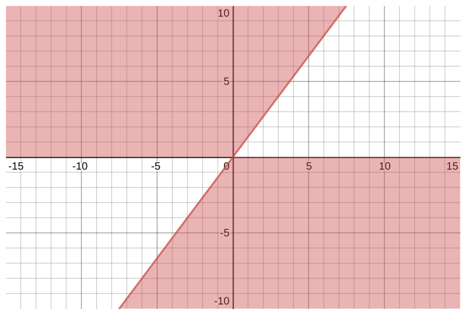
can be expressed as

$$x_1^2 - 4x_1x_2 + 4x_2^2 \geq x_1^2 + 2x_1x_2 + x_2^2,$$

and thus

$$x_2(4x_1 - 3x_2) \leq 0.$$

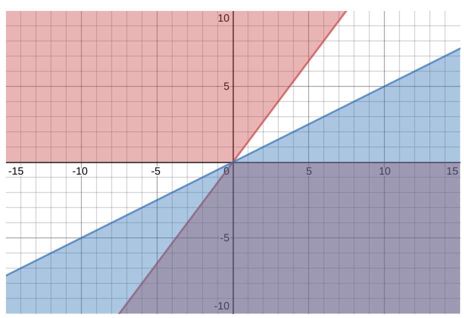
Plotting this region (*red*), we can immediately see that it is not convex:



and thus cannot possibly be equivalent to the first, unsquared constraint.

(b) What additional constraint must be imposed alongside the squared constraint to enforce the same feasible set as the unsquared SOC constraint?

Solution: The initial SOC constraint $x_1 - 2x_2 \geq \|\vec{x}\|_2$ implicitly imposes that $x_1 - 2x_2 \geq 0$, a condition that is lost when both sides of the inequality are squared. We can recover the same feasible set (*purple*) by imposing this constraint (*blue*) alongside $(x_1 - 2x_2)^2 \geq \|\vec{x}\|_2^2$ (*red*):



This principle holds in general; it is indeed possible to square an SOC constraint

$$\|A\vec{x} + \vec{b}\|_2 \leq \vec{c}^\top \vec{x} + d,$$

provided one takes care to include the implicit constraint that $\vec{c}^\top \vec{x} + d \geq 0$. The general SOC constraint is thus equivalent to

$$\begin{aligned} \|A\vec{x} + \vec{b}\|_2^2 &\leq (\vec{c}^\top \vec{x} + d)^2 \\ 0 &\leq \vec{c}^\top \vec{x} + d. \end{aligned}$$

Note, however, that this is often of limited use, as certain convex optimization solvers (e.g., CVX) may not accept constraints of the form $\|A\vec{x} + \vec{b}\|_2^2 - (\vec{c}^\top \vec{x} + d)^2 \leq 0$ in general, since, as shown above, the difference of two convex quadratic functions may well be nonconvex. Although you may know that your problem is convex — i.e., that you've provided the requisite additional constraints — your solver might not be that smart!

3. Casting optimization problems as SOCPs

Cast the following problem as an SOCP in its standard form:

$$\min_{\vec{x}\in\mathbb{R}^n} \sum_{i=1}^{p} \frac{\|F_i\vec{x}+\vec{g}_i\|_2^2}{\vec{a}_i^\top \vec{x}+b_i}$$
$$\text{s.t. } \vec{a}_i^\top \vec{x}+b_i > 0, \quad i=1,\ldots,p,$$

where $F_i \in \mathbb{R}^{m\times n}$, $\vec{g}_i \in \mathbb{R}^m$, $\vec{a}_i \in \mathbb{R}^n$, and $b_i \in \mathbb{R}$, for $i=1,\ldots,p$.

Solution: Introducing slack variables, the problem can be written as

$$\begin{aligned} \min_{\vec{x}\in\mathbb{R}^m,\ \vec{t}\in\mathbb{R}^p} \ & \sum_{i=1}^{p} t_i \\ \text{s.t. } & (F_i\vec{x}+\vec{g}_i)^\top (F_i\vec{x}+\vec{g}_i) \le t_i(\vec{a}_i^\top\vec{x}+b_i), \quad i=1,\ldots,p \\ & \vec{a}_i^\top\vec{x}+b_i > 0, \quad i=1,\ldots,p, \end{aligned}$$

or equivalently,

$$\begin{aligned} \min_{\vec{x}\in\mathbb{R}^m,\ \vec{t}\in\mathbb{R}^p} \ & \sum_{i=1}^{p} t_i \\ \text{s.t. } & \left\| \begin{bmatrix} 2(F_i\vec{x}+\vec{g}_i) \\ t_i-\vec{a}_i^\top\vec{x}-b_i \end{bmatrix} \right\|_2 \le t_i+\vec{a}_i^\top\vec{x}+b_i, \quad i=1,\ldots,p, \end{aligned}$$

which is an SOCP. For a full discussion of this equivalence, see Calafiore & El Ghaoui section 10.2.3.

4. A review of standard problem formulations

In this question, we review conceptually the standard forms of various problems and the assertions we can (and cannot!) make about each.

(a) ***Linear programming (LP).***

i. Write the most general form of a linear program (LP) and list its defining attributes.

Solution: A general LP can be written as

$$\begin{aligned} p^* = \min_{\vec{x}\in\mathbb{R}^n} \ & \vec{c}^\top\vec{x}+d \\ \text{s.t. } & A_{\text{eq}}\vec{x} = \vec{b}_{\text{eq}} \\ & A\vec{x} \le \vec{b}, \end{aligned}$$

or equivalently,

$$\begin{aligned} p^* = \min_{\vec{x}\in\mathbb{R}^n} \ & \vec{c}^\top\vec{x}+d \\ \text{s.t. } & A_{\text{eq}}\vec{x} = \vec{b}_{\text{eq}} \\ & \vec{x} \ge \vec{0}. \end{aligned}$$

The first LP formulation is known as the *inequality form*; the second is known as the *conic form*. A full treatment of the equivalence of these forms and how to convert between them can be found in section 9.3 of Calafiore & El Ghaoui.

ii. Under what conditions is an LP convex?
Solution: An LP is **always convex**, as the objective function and all constraints are convex (affine) and all equality constraints are affine.

(b) ***Quadratic programming (QP).***

i. Write the most general form of a quadratic program (QP) and list its defining attributes.
Solution: A general QP can be written as

$$\begin{aligned} p^* = \min_{\vec{x}\in\mathbb{R}^n} \ & \frac{1}{2}\vec{x}^\top H\vec{x} + \vec{c}^\top\vec{x} + d \\ \text{s.t. } & A_{\text{eq}}\vec{x} = \vec{b}_{\text{eq}} \\ & A\vec{x} \leq \vec{b}. \end{aligned}$$

ii. Under what conditions is a QP convex?
Solution: A QP is convex if and only if $H \succeq 0$ (i.e., PSD).

(c) ***Quadratically-constrained quadratic programming (QCQP).***

i. Write the most general form of a quadratically-constrained quadratic program (QCQP) and list its defining attributes.
Solution: A general QCQP can be written as

$$\begin{aligned} p^* = \min_{\vec{x}\in\mathbb{R}^n} \ & \vec{x}^\top H_0\vec{x} + 2\vec{c}^\top\vec{x} + d \\ \text{s.t. } & \vec{x}^\top H_i\vec{x} + 2\vec{c}_i^\top\vec{x} + d_i \leq 0, \quad i = 1,\ldots,m \\ & \vec{x}^\top H_j\vec{x} + 2\vec{c}_j^\top\vec{x} + d_j = 0, \quad j = 1,\ldots,q. \end{aligned}$$

ii. Under what conditions is a QCQP convex?
Solution: A QCQP is convex if and only if all matrices H_0 and H_i, $i = 1,\ldots,m$ are PSD, and $H_j = 0$ for all $j = 1,\ldots,q$ (i.e., when the objective and all inequality constraints are convex quadratic, and all the equality constraints are actually affine.

(d) ***Second-order cone programming (SOCP).***

i. Write the most general form of a second-order cone program (SOCP) and list its defining attributes.
Solution: A general SOCP can be written as

$$\begin{aligned} p^* = \min_{\vec{x}\in\mathbb{R}^n} \ & \vec{c}^\top\vec{x} \\ \text{s.t. } & \|A_i\vec{x} + \vec{b}_i\|_2 \leq \vec{c}_i^\top\vec{x} + d_i, \quad i = 1,\ldots,m, \end{aligned}$$

or equivalently,

$$\begin{aligned} p^* = \min_{\vec{x}\in\mathbb{R}^n} \ & \vec{c}^\top\vec{x} \\ \text{s.t. } & (A_i\vec{x} + \vec{b}_i, \vec{c}_i^\top\vec{x} + d_i) \in \mathcal{K}_{m_i} \quad i = 1,\ldots,m, \end{aligned}$$

where second-order cone (SOC) $\mathcal{K}_n \doteq \{(\vec{x}, t), \vec{x} \in \mathbb{R}^n, t \in \mathbb{R} \mid \|\vec{x}\|_2 \leq t\}$. The first SOCP formulation is known as the *standard inequality form*; the second is known as the *conic standard form*.

ii. Under what conditions is an SOCP convex?
Solution: An SOCP is **always convex**, as the objective function is convex (linear) and the constraint stipulates that points lie within a convex set. A thorough discussion of the convexity of SOC constraints can be found in Calafiore & El Ghaoui chapter 10.

(e) ***Relationships.*** Recall that

$$LP \subset QP_{\text{convex}} \subset QCQP_{\text{convex}} \subset SOCP \subset \{\text{all convex programs}\},$$

where LP denotes the set of all linear programs, QP_{convex} denotes the set of all convex quadratic programs, etc. Which of these problems can be solved most efficiently? Why are these categorizations useful?

Solution: In general, problems to the left side of the subset sequence above can be solved more efficiently than those on the right, for problems of comparable size: LPs are arguably easiest (optima are always achieved at a critical point, so we just need to check those analytically, e.g. using the simplex algorithm), while general convex problems must use numerical algorithms like gradient descent and Newton's method, whose efficiency depends on the particular geometric characteristics of the problem. Intermediate forms (convex QPs/QCQPs, SOCPs) fall somewhere in between in terms of difficulty: there's lots of research on how to solve them efficiently (one popular set of approaches is called "interior point methods"), and there are a number of off-the-shelf solvers available (e.g., CVX).

Though we don't get a chance to explore it much in this class, knowing these classes of problems is useful when encountering optimization problems in the wild — if you can write your problem in one of the forms above, you know what kinds of solutions and convergence guarantees you can expect, and often employ existing software to help you solve it. For further discussion of these problem classes — as well as an even more general class known as semidefinite programming (SDP), of which SOCPs are a subset — we encourage you to take EECS 227B and 227C.