COMPUTATIONAL METHODS IN OPTIMIZATION

November 20, 2023

Today: SOCP

Tomorrow: Lab/Tutorial on conic programs

Coming up: Last two lectures on implementation challenges

NB: Revised version on November 21
(fixed incorrect reformulation on page 5)

SOCP (Second-Order Cone Programming)

↳ Largest class of (convex!) optimization problems we can apply IPMs to with theoretical guarantees: Conic Programming

Conic program (in standard form)

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad c^T x \quad \text{s.t.} \quad \begin{array}{l} Ax=b \\ x\in K \end{array}$$

where $c\in\mathbb{R}^n$, $A\in\mathbb{R}^{m\times n}$, $b\in\mathbb{R}^m$ and K is a closed, convex cone

$$\left(\forall x\in K, \forall t\geq 0, \ tx\in K\right)$$

Ex) $K=\{x\in\mathbb{R}^n \mid x\geq 0\}$: linear programming

$K=\{X\in\mathbb{R}^{\sqrt{n}\times\sqrt{n}} \mid X\succeq 0\}$: SDP (with x being a vectorization of X)

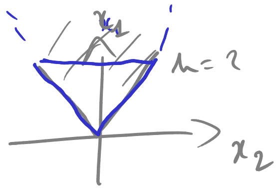
⋮

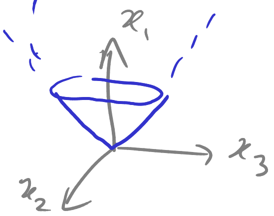
many more!

Our focus today: Second-order cones

Def: The "ice cream cone" in $\mathbb{R}^n$ (also called the quadratic cone/the second-order cone) is the set

$$K_q=\{x\in\mathbb{R}^n \mid x_1^2\geq\sum_{j=2}^n x_j^2, \ x_1\geq 0\}$$





The rotated quadratic cone is

$$K_r = \{ x \in \mathbb{R}^n \mid \underset{\uparrow}{2x_1x_2} \geq \sum_{j=3}^{n} \underset{\uparrow}{x_j^2}, \; x_1 \geq 0, x_2 \geq 0 \}$$

degree 2 in $(x_1, - x_n)$

→ K_q and K_r are the most common second-order cones

→ More generally, a closed convex cone K is second-order if it can be written as

$$\{ x \in \mathbb{R}^n \mid \|A_i x + b_i\| \leq c_i^T x + d_i \quad i = 1..m \}$$

where $A_i \in \mathbb{R}^{m_i \times n}$, $b_i \in \mathbb{R}^{m_i}$, $c_i \in \mathbb{R}^n$, $d_i \in \mathbb{R}$

NB: $\|A_i x + b_i\| \leq c_i^T x + d_i \iff \|A_i x_i + b_i\|^2 \leq (c_i^T x + d_i)^2$
$c_i^T x + d_i \geq 0$

Definition: A second-order cone program (SOCP) in standard form is given by

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad c^T x \quad \text{s.t.} \quad \|A_i x + b_i\| \leq c_i^T x + d_i \quad i = 1..m$$

where $c \in \mathbb{R}^n$, $\underbrace{A_i \in \mathbb{R}^{m_i \times n}, \; b_i \in \mathbb{R}^{m_i}, \; c_i \in \mathbb{R}^n, \; d_i \in \mathbb{R}}_{\forall i = 1..m}$

- SOCP generalizes LP

$$\text{(LP)} \quad \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad c^T x \quad \text{s.t.} \quad a_i^T x = b_i \quad \forall i = 1..m$$
$$x_j \geq 0 \quad \forall j = 1..n$$

$$a_i^T x - b_i = 0 \iff \begin{array}{l} 0 \leq a_i^T x - b_i \\ 0 \leq -a_i^T x + b_i \end{array}$$

$0 \le a_i^T x - b_i$ is a second-order cone constraint of the form $\|\bar{A}x + \bar{b}\| \le \bar{c}^T x + \bar{d}$

where $\bar{A} = 0_{\mathbb{R}^{m\times n}}$, $\bar{b} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^m$

$\bar{c} = a_i$ and $\bar{d} = -b_i$

$0 \le a_i^T x - b_i$

$0 \le -a_i^T x + b_i$

$\bar{A} = \begin{bmatrix} -a_i^T \\ +a_i^T \end{bmatrix}$ $\bar{b} = \begin{bmatrix} b_i \\ -b_i \end{bmatrix}$

$\bar{c} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$ $\bar{d} = 0$

$\|+a_i^T x + b_i\| \le 0$ $\bar{A} = \begin{bmatrix} -a_i^T \end{bmatrix}$ $\bar{b} = \begin{bmatrix} b_i \end{bmatrix}$

$0 \le x_j$ $\bar{A} = 0$ $\bar{b} = 0$ $\bar{c} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ $\bar{d} = 0$

- SOCP generalizes QP

$$(QP) \quad \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad c^T x + \frac{1}{2} x^T Q x \qquad Ax = b, \quad x \ge 0$$

$$Q \succeq 0 \quad (Q = Q^T)$$

$Ax = b$, $x \ge 0$ can be reformulated as for (LP)

To reformulate the objective, we introduce additional variables $\underbrace{w \in \mathbb{R}^n}_{w^T w = x^T Q x}$, $u \in \mathbb{R}$, $\underbrace{v \in \mathbb{R}}_{x^T Q x}$, so that the problem

$\Uparrow$

$w = Q^{1/2} x$

becomes

$$\begin{cases} \underset{\substack{x \in \mathbb{R}^n \\ v \in \mathbb{R} \\ w \in \mathbb{R}^m \\ u \in \mathbb{R}}}{\text{minimize}} & c^T x + \frac{1}{2} v \\ \text{s.t.} & \{ \quad \} \leftarrow \begin{matrix} Ax=b \\ x \geq 0 \end{matrix} \\ & w^T w \leq u v \\ & u = 1 \\ & w = Q^{1/2} x \end{cases}$$

required so that the inequality constraint is conic

↳ A further reformulation gets rid of w and u (see Exercise 4 in lab session)

Remark: An SOCP solver / conic programming solver like Scip or MOSEK accepts SOCP reformulations of LPs and QPs but can identify that simpler formulations exist (in which case it will solve the simpler formulation using an LP or QP solver)

Solving SOCPs

- For conic programs, the dual of $\underset{x \in \mathbb{R}^n}{\text{minimize}}\ c^T x$ s.t. $\overset{\in \mathbb{R}^{m \times n}}{A} x = b$, $x \in K$

is $\underset{\substack{y \in \mathbb{R}^m \\ s \in \mathbb{R}^n}}{\text{maximize}}\ b^T y$ s.t. $A^T y + s = c$, $s \in K^*$

In IPMs, we use $\mu = \frac{x^T s}{n}$ as convergence criterion

K^* dual cone of K $\quad K^\circ = \{ s \in \mathbb{R}^n \mid x^T s \geq 0 \ \forall x \in K \}$

$\Rightarrow$ IPMs compute primal-dual steps by solving the optimality conditions $Ax = b$ $A^T y + s = c$

$x \in K$ $s \in K^*$

$x_i s_i = 0$ $\forall i = 1..n$

Strict feasibility: $x \in \text{int}(K)$ $\text{int}(K)$ = interior of the cone

$s \in \text{int}(K^*)$

↳ For SOCPs,

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad c^T x \quad \text{s.t.} \quad \|A_i x + b_i\| \leq c_i^T x + d_i \quad i = 1..m$$

$A_i \in \mathbb{R}^{m_i \times n}$

the dual is

$$\text{maximize} \left(\sum_{i=1}^{m} b^T y_i - z_i d_i \right) \quad \text{s.t.} \quad \sum_{i=1}^{m} (A_i^T y_i - z_i c_i) = c$$

$$\underbrace{\left(\sum_{i=1}^{m} m_i \right)}_{\uparrow \; y_1, .., y_m} + \underbrace{m}_{\uparrow \; z} \quad \begin{cases} y_i \in \mathbb{R}^{m_i} \\ z \in \mathbb{R}^m \end{cases}$$

$\|y_i\| \leq z_i$

$\begin{bmatrix} z_i \\ y_i \end{bmatrix} \in \mathbb{R}^{m_i+1}$ belongs to the ice cream cone in $\mathbb{R}^{m_i+1}$

(The dual of an SOCP is also an SOCP)

Optimality conditions: $\|A_i x + b_i\| \leq c_i^T x + d_i \iff \begin{bmatrix} c_i^T x + d_i \\ A_i x + b_i \end{bmatrix} \in K_q(\mathbb{R}^{m_i+1})$

$$\sum_{i=1}^{m} (A_i^T y_i - z_i c_i) = c$$

$$\|y_i\| \leq z_i \iff \begin{bmatrix} z_i \\ y_i \end{bmatrix} \in K_q(\mathbb{R}^{m_i+1})$$

$\forall i = 1..m$, $[A_i x + b_i]_j [y_i]_j = 0$ $\forall j = 1..m_i$

$(c_i^T x + d_i) z_i = 0$

↳ Complementarity for the vectors $\begin{bmatrix} c_i^T x + d_i \\ A_i x + b_i \end{bmatrix}$ and $\begin{bmatrix} z_i \\ y_i \end{bmatrix}$

Application : Robust LP

↳ SOCPs are very useful to model uncertainty

↳ Suppose that we want to solve the LP

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad c^T x \quad \text{s.t.} \quad \begin{array}{l} Ax = b \iff a_i^T x = b_i \quad \forall i = 1..m \\ x \geq 0 \end{array}$$

but the vectors a_i are uncertain but they belong to a known set

$$\forall i = 1..m, \quad a_i \in \mathcal{E}_i = \{ \bar{a}_i + P_i u, \|u\| \leq 1 \quad P_i = P_i^T \succeq 0 \}$$

$$\bar{a}_i \in \mathbb{R}^n$$

$\mathcal{E}_i$: ellipsoidal uncertainty set

Goal: Compute the best x regardless of the value of the a_i's ("robust to the value of a_i")

Since $\max\{ a_i^T x : a_i \in \mathcal{E}_i \} = \bar{a}_i^T x + \|P_i x\|$,

we can write a robust version of the LP as

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad c^T x \quad \text{s.t.} \quad \begin{array}{l} \bar{a}_i^T x + \|P_i x\| \leq b_i \\ -\bar{a}_i^T x + \|P_i x\| \leq b_i \\ -x_j \leq 0 \qquad \forall j = 1..m \end{array}$$

⇒ this is no longer an LP but it is an SOCP