

Quadratically constrained quadratic  
programming,  
Second-order cone programming  
Computational Intelligence, Lecture 9

by Sergei Savin

Spring 2021

- Quadratic programming: recap
- Quadratically constrained quadratic programming
  - ▶ General form
  - ▶ Domain
- Second-order cone programming
  - ▶ General form
  - ▶ Special cases
  - ▶ Friction cone
  - ▶ Friction cone, solution
- Homework

# QUADRATIC PROGRAMMING

## General form

Remember the general form of a quadratic program:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && \mathbf{x}^\top \mathbf{H} \mathbf{x} + \mathbf{f}^\top \mathbf{x}, \\ & \text{subject to} && \begin{cases} \mathbf{A} \mathbf{x} \leq \mathbf{b}, \\ \mathbf{F} \mathbf{x} = \mathbf{g}. \end{cases} \end{aligned} \tag{1}$$

where  $\mathbf{H}$  is positive-definite and  $\mathbf{A} \mathbf{x} \leq \mathbf{b}$  describe a convex region.

# QUADRATICALLY CONSTRAINED QUADRATIC PROGRAMMING

## General form

General form of a quadratically constrained quadratic program (QCQP) is given below:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && \mathbf{x}^\top \mathbf{P}_0 \mathbf{x} + \mathbf{q}_0^\top \mathbf{x}, \\ & \text{subject to} && \begin{cases} \mathbf{x}^\top \mathbf{P}_i \mathbf{x} + \mathbf{q}_i^\top \mathbf{x} + r_i \leq 0, \\ \mathbf{F} \mathbf{x} = \mathbf{g}. \end{cases} \end{aligned} \tag{2}$$

where  $\mathbf{P}_i$  are positive-definite.

# QUADRATICALLY CONSTRAINED QUADRATIC PROGRAMMING

## Domain

Domain of a QCQP without equality constraints and with no degenerate inequality constraints is an intersection of ellipses:



Set  $\mathbf{P}_i = \mathbf{0}$  and you get a QP.

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && \mathbf{x}^\top \mathbf{P}_0 \mathbf{x} + \mathbf{q}_0^\top \mathbf{x}, \\ & \text{subject to} && \begin{cases} \begin{bmatrix} \mathbf{q}_1^\top \\ \dots \\ \mathbf{q}_n^\top \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} r_1 \\ \dots \\ r_n \end{bmatrix} \\ \mathbf{F} \mathbf{x} = \mathbf{g}. \end{cases} \end{aligned} \tag{3}$$

Set  $\mathbf{P}_0 = \mathbf{0}$  and you get an LP.

# SECOND-ORDER CONE PROGRAMMING

## General form

The general form of a Second-order cone program (SOCP) is:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && \mathbf{f}^\top \mathbf{x}, \\ & \text{subject to} && \begin{cases} \|\mathbf{A}_i \mathbf{x} + \mathbf{b}_i\|_2 \leq \mathbf{c}_i^\top \mathbf{x} + d_i, \\ \mathbf{F} \mathbf{x} = \mathbf{g}. \end{cases} \end{aligned} \tag{4}$$

LP, QP and QCQP are subsets of SOCP.

# SECOND-ORDER CONE PROGRAMMING

## Special cases

We can write problem where our domain is a ball as SOCP:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && \mathbf{f}^\top \mathbf{x}, \\ & \text{subject to} && \|\mathbf{x}\|_2 \leq d_i \end{aligned} \tag{5}$$

Same for ellipsoidal constraints:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && \mathbf{f}^\top \mathbf{x}, \\ & \text{subject to} && \|\mathbf{A}_i \mathbf{x}\|_2 \leq d_i \end{aligned} \tag{6}$$



# SOCP TO QCQP

## Part 1

Set  $\mathbf{c}_i = 0$  and  $d_i = 0$  and recognize that  $\|\mathbf{A}_i \mathbf{x} + \mathbf{b}_i\|_2 \leq 0$  is the same as  $(\mathbf{A}_i \mathbf{x} + \mathbf{b}_i)^\top (\mathbf{A}_i \mathbf{x} + \mathbf{b}_i) \leq 0$

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & \mathbf{f}^\top \mathbf{x}, \\ \text{subject to} & \begin{cases} \mathbf{x}^\top \mathbf{A}_i^\top \mathbf{A}_i \mathbf{x} + 2\mathbf{b}_i^\top \mathbf{A}_i \mathbf{x} + \mathbf{b}_i^\top \mathbf{b}_i \leq 0 \\ \mathbf{F} \mathbf{x} = \mathbf{g}. \end{cases} \end{array} \quad (7)$$

# SOCP TO QCQP

## Part 2

Now to make the cost quadratic:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && t, \\ & \text{subject to} && \begin{cases} \mathbf{x}^\top \mathbf{A}_0^\top \mathbf{A}_0 \mathbf{x} + 2\mathbf{b}_0^\top \mathbf{A}_0 \mathbf{x} + \mathbf{b}_0^\top \mathbf{b}_0 \leq t \\ \mathbf{x}^\top \mathbf{A}_i^\top \mathbf{A}_i \mathbf{x} + 2\mathbf{b}_i^\top \mathbf{A}_i \mathbf{x} + \mathbf{b}_i^\top \mathbf{b}_i \leq 0 \\ \mathbf{F}\mathbf{x} = \mathbf{g}. \end{cases} \end{aligned} \quad (8)$$

Which is the same as:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && \mathbf{x}^\top \mathbf{H}\mathbf{x} + \mathbf{f}^\top \mathbf{x}, \\ & \text{subject to} && \begin{cases} \mathbf{x}^\top \mathbf{A}_i^\top \mathbf{A}_i \mathbf{x} + 2\mathbf{b}_i^\top \mathbf{A}_i \mathbf{x} + \mathbf{b}_i^\top \mathbf{b}_i \leq 0 \\ \mathbf{F}\mathbf{x} = \mathbf{g}. \end{cases} \end{aligned} \quad (9)$$

As long as  $\mathbf{A}_0 = \sqrt{\mathbf{H}}$ , and  $\mathbf{b}_0 = 0.5\mathbf{A}_0^{-1}\mathbf{f}$ .

# FRICTION CONE

## Friction and normal reaction force relation



Let  $\mathbf{f}$  be total reaction force,  $\mathbf{f}_n$  be its normal component (perpendicular to the surface locally), also known as normal reaction; and let  $\mathbf{f}_{fr}$  be its tangential component (a vector lying in the tangent plane to the surface, constructed at the contact point), or friction force. Let  $\mathbf{e}_n$  be a unit vector, normal to the surface at the point of contact.

$$\mathbf{f} = \mathbf{f}_n + \mathbf{f}_{fr} \quad (10)$$

# SECOND-ORDER CONE PROGRAMMING

## Friction cone

Defining  $\mathbf{E}_t = [\mathbf{e}_{t,1}, \mathbf{e}_{t,2}] = \mathcal{L}(\mathbf{e}_n)$  be an orthonormal basis in the tangential space to the surface, we can write:

$$\mathbf{f} = \mathbf{e}_n n + \mathbf{E}_t \mathbf{t}$$

$$\mathbf{f}_n = \mathbf{e}_n n$$

$$\mathbf{f}_{fr} = \mathbf{E}_t \mathbf{t}$$

$$\mathbf{t} = [t_1, t_2]$$

The friction cone conditions could be written in any of the following ways:

$$\sqrt{t_1^2 + t_2^2} < \mu n \quad (11)$$

$$\|\mathbf{E}_t^\top \mathbf{f}\| \leq \mu \mathbf{e}_n^\top \mathbf{f} \quad (12)$$

where  $\mu$  is a friction coefficient.

Implement a program that finds right-most point of an intersection of two ellipsoids; visualise the problem and the solution.

Lecture slides are available via Moodle.

You can help improve these slides at:

[github.com/SergeiSa/Computational-Intelligence-Slides-Spring-2021](https://github.com/SergeiSa/Computational-Intelligence-Slides-Spring-2021)



Check Moodle for additional links, videos, textbook suggestions.