

Quadratically constrained quadratic
programming,
Second-order cone programming
Computational Intelligence, Lecture 8

by Sergei Savin

Fall 2020

- 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:

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

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:

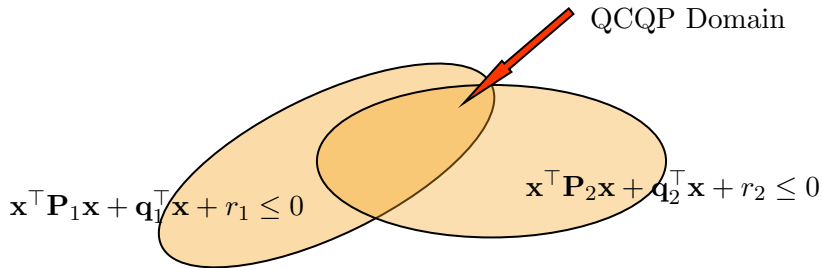
$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && \mathbf{x}^\top \mathbf{P}_0 \mathbf{x} + \mathbf{q}_0^\top \mathbf{x}, \\ & \text{subject to} && \begin{cases} \mathbf{x}^\top \mathbf{P}_i \mathbf{x} + \mathbf{q}_i^\top \mathbf{x} + r_i \leq 0, \\ \mathbf{A} \mathbf{x} = \mathbf{b}. \end{cases}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

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:



Second-order cone programming

General form

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

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && \mathbf{f}^\top \mathbf{x}, \\ & \text{subject to} && \begin{cases} \|\mathbf{A}_i \mathbf{x} + \mathbf{b}_i\|_2 \leq \mathbf{c}_i^\top \mathbf{x} + d_i, \\ \mathbf{F} \mathbf{x} = g. \end{cases} \end{aligned} \tag{3}$$

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:

$$\begin{array}{ll}\underset{\mathbf{x}}{\text{minimize}} & \mathbf{f}^\top \mathbf{x}, \\ \text{subject to} & \|\mathbf{x}\|_2 \leq d_i\end{array}\tag{4}$$

Same for ellipsoidal constraints:

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

Second-order cone programming

Friction cone

Remember *friction cone*: the idea that for a reaction force should lie in a cone normal to the ground.

Let $\mathbf{f} = \mathbf{e}_n n + \mathbf{e}_{t,1} t_1 + \mathbf{e}_{t,2} t_2$ where \mathbf{e}_n is a normal direction to the ground, and $\mathbf{e}_{t,i}$ are tangential directions (all unit vectors). Therefore $\mathbf{e}_n n$ is the normal reaction and $\mathbf{e}_{t,1} t_1 + \mathbf{e}_{t,2} t_2$ is the friction force.

The friction cone conditions could be written as:

$$\sqrt{t_1^2 + t_2^2} < \mu n \quad (6)$$

where μ is a friction coefficient.

Second-order cone programming

Friction cone, solution

We can represent condition (6) in the SOC form, taking \mathbf{f} as our variable:

$$\| [\mathbf{e}_{t,1} \quad \mathbf{e}_{t,2}]^\top \mathbf{f} \| \leq \mu \mathbf{e}_n^\top \mathbf{f} \quad (7)$$

If we want to impose strict inequality, we should add a margin on μ , proposing to use $\mu^* < \mu$ instead.

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-Fall-2020

Check Moodle for additional links, videos, textbook suggestions.