Quadratic programming

Computational Intelligence, Lecture 7

by Sergei Savin

Spring 2025

Content

- Quadratic programming
- Quadratically constrained quadratic programming (QCQP)
  - QCQP to LP; QCQP to LP
  - Ellipsoidal constraints $\rightarrow$ to canonical form

Positive-definite matrices the following properties:

Eigenvalues

All eigenvalues of a positive-definite (PD) matrix are real and positive.

Eigenvectors

Eigenvectors of a PD matrix with form an orthogonal basis.

Proof (for the case of distinct eigenvalues): let $\mathbf{M}$ be a PD matrix, and $\mathbf{M}\mathbf{v} = \lambda \mathbf{v}$ and $\mathbf{M}\mathbf{u} = \gamma \mathbf{u}$. Then:

$$\mathbf{u}^\top \mathbf{M} \mathbf{v} = \mathbf{u}^\top (\mathbf{M}\mathbf{v}) = \lambda \mathbf{u}^\top \mathbf{v} \tag{1}$$

$$\mathbf{u}^\top \mathbf{M} \mathbf{v} = (\mathbf{u}^\top \mathbf{M})\mathbf{v} = \gamma \mathbf{u}^\top \mathbf{v} \tag{2}$$

Therefore, $\lambda \mathbf{u}^\top \mathbf{v} = \gamma \mathbf{u}^\top \mathbf{v}$, and since λ and γ are eigenvalues and and eigenvalues are distinct, so $\lambda \neq \gamma$. This implies that $\mathbf{u}^\top \mathbf{v} = 0$.

Remember the general form of a quadratic program:

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

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

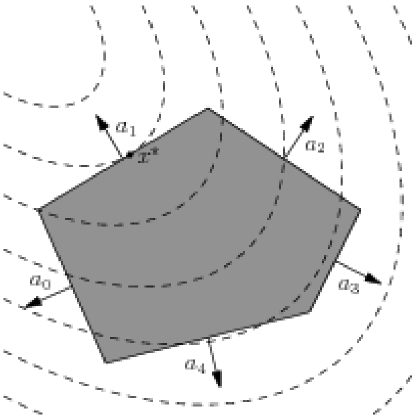


Figure 1: Geometry of a QP. Source

The cost function of a QP has the form $c(\mathbf{x}) = \mathbf{x}^\top \mathbf{H}\mathbf{x} + \mathbf{f}^\top \mathbf{x}$. Let us show that the requirement that $\mathbf{H}$ is positive-definite does not limit the range of convex problems that can be solved as a QP.

Let $\mathbf{M}$ be a non-symmetric matrix. Quadratic form $q(\mathbf{x}) = \mathbf{x}^\top \mathbf{M}\mathbf{x}$ is a scalar and is equal to its transpose:

$$q(\mathbf{x}) = \mathbf{x}^\top \mathbf{M}\mathbf{x} \tag{4}$$

$$q(\mathbf{x}) = 0.5(\mathbf{x}^\top \mathbf{M}\mathbf{x} + \mathbf{x}^\top \mathbf{M}\mathbf{x}) \tag{5}$$

$$q(\mathbf{x}) = 0.5(\mathbf{x}^\top \mathbf{M}\mathbf{x} + \mathbf{x}^\top \mathbf{M}^\top \mathbf{x}) \tag{6}$$

$$q(\mathbf{x}) = 0.5\mathbf{x}^\top (\mathbf{M} + \mathbf{M}^\top)\mathbf{x} \tag{7}$$

Equivalently prove that the cost function $c(\mathbf{x})$ is always equivalent to the cost function $c(\mathbf{x}) = 0.5\mathbf{x}^\top (\mathbf{H} + \mathbf{H}^\top)\mathbf{x} + \mathbf{f}^\top \mathbf{x}$. Because of that, without a loss of generality we can assume $\mathbf{H}$ to be symmetric.

Let us prove that $\mathbf{H}$ needs to be positive semi-definite in order for $c(\mathbf{x})$ to be convex.

Assume that one of the eigenvalues of $\mathbf{H}$ is negative: $\mathbf{H}\mathbf{v} = \lambda \mathbf{v}$, where $\lambda < 0$ and $||\mathbf{v}|| = 1$. We can find values of $c(0)$, $c(\mathbf{v})$ and $c(2\mathbf{v})$:

$$c(0) = 0 \tag{8}$$

$$c(\mathbf{v}) = \mathbf{v}^\top \mathbf{H}\mathbf{v} = \lambda \mathbf{v}^\top \mathbf{v} = \lambda \tag{9}$$

Note that $c((1-\beta)0 + \beta\mathbf{v}) = c(\beta\mathbf{v}) = \lambda\beta^2$ and $(1-\beta)c(0) + \beta c(\mathbf{v}) = \lambda\beta$. Since $0 < \beta < 1$ we conclude $\beta^2 < \beta$; since $\lambda < 0$, $\lambda\beta^2 > \lambda\beta$. So $c((1-\beta)0 + \beta\mathbf{v}) > (1-\beta)c(0) + \beta c(\mathbf{v})$. Thus, such $c(\mathbf{x})$ is not convex. $\square$

Let us consider a QP with degenerate matrix $\mathbf{H} = \mathbf{R}\Sigma\mathbf{R}^\top$ (without a loss of generality $\mathbf{H}$ can be assumed to be symmetric), where $\mathbf{R} \in \mathbb{R}^{n \times k}$.

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{x}^\top \mathbf{R}\Sigma\mathbf{R}^\top \mathbf{x} + \mathbf{f}^\top \mathbf{x}, \\ \text{subject to} \quad & \begin{cases} \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\ \mathbf{F}\mathbf{x} = \mathbf{g}. \end{cases} \end{aligned} \tag{10}$$

Matrix $\mathbf{R}$ is a row-space basis of $\mathbf{H}$, and $\mathbf{N}$ is its orthogonal compliment. Then we can decompose $\mathbf{x}$ as $\mathbf{x} = \mathbf{R}\zeta + \mathbf{N}\mathbf{z}$:

$$\begin{aligned} \underset{\zeta, \mathbf{z}}{\text{minimize}} \quad & \zeta^\top \Sigma \zeta + \mathbf{f}^\top \mathbf{R}\zeta + \mathbf{f}^\top \mathbf{N}\mathbf{z}, \\ \text{subject to} \quad & \begin{cases} \mathbf{A}\mathbf{R}\zeta + \mathbf{A}\mathbf{N}\mathbf{z} \leq \mathbf{b}, \\ \mathbf{F}\mathbf{R}\zeta + \mathbf{F}\mathbf{N}\mathbf{z} = \mathbf{g}. \end{cases} \end{aligned} \tag{11}$$

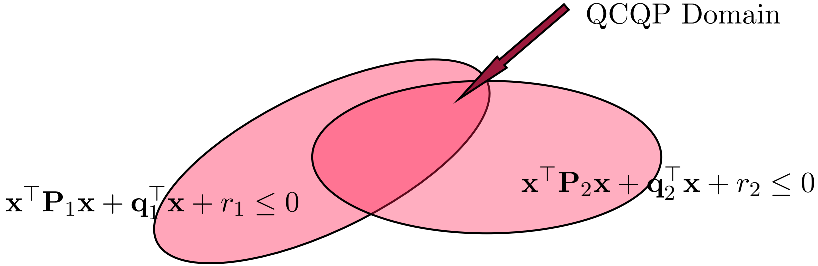
General form of a quadratically constrained quadratic program (QCQP) is given below:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{x}^\top \mathbf{P}_0 \mathbf{x} + \mathbf{q}_0^\top \mathbf{x}, \\ \text{subject to} \quad & \begin{cases} \mathbf{x}^\top \mathbf{P}_i \mathbf{x} + \mathbf{q}_i^\top \mathbf{x} + r_i \leq 0, \\ \mathbf{F}\mathbf{x} = \mathbf{g}. \end{cases} \end{aligned} \tag{12}$$

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

QCQP - 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.

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} \quad && \mathbf{x}^\top \mathbf{P}_0 \mathbf{x} + \mathbf{q}_0^\top \mathbf{x}, \\ &\text{subject to} && \begin{cases} \begin{bmatrix} \mathbf{q}_1^\top \\ \dots \\ \mathbf{q}_n^\top \end{bmatrix} \mathbf{x} \le \begin{bmatrix} -r_1 \\ \dots \\ -r_n \end{bmatrix} \\ \mathbf{F}\mathbf{x} = \mathbf{g}. \end{cases} \end{aligned} \tag{13}$$

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

Can we re-write the expression $\mathbf{x}^\top \mathbf{P} \mathbf{x} + \mathbf{q}^\top \mathbf{x} + r \leq 0$ as a canonical form ellipsoid:

$$\frac{z_1^2}{m_1^2} + \frac{z_2^2}{m_2^2} + ... + \frac{z_n^2}{m_n^2} \leq 1 \tag{14}$$

We start by proposing a substitution $\mathbf{x}_0 = -\frac{1}{2}\mathbf{P}^{-1}\mathbf{q}$ and $-d = r - \mathbf{x}_0^\top \mathbf{P} \mathbf{x}_0$. We can prove that:
$(\mathbf{x} - \mathbf{x}_0)^\top \mathbf{P} (\mathbf{x} - \mathbf{x}_0) - d = \mathbf{x}^\top \mathbf{P} \mathbf{x} + \mathbf{q}^\top \mathbf{x} + r$

$$(\mathbf{x} - \mathbf{x}_0)^\top \mathbf{P} (\mathbf{x} - \mathbf{x}_0) - d = \tag{15}$$

$$= \mathbf{x}^\top \mathbf{P} \mathbf{x} - 2\mathbf{x}_0^\top \mathbf{P} \mathbf{x} + \mathbf{x}_0^\top \mathbf{P} \mathbf{x}_0 - d = \tag{16}$$

$$= \mathbf{x}^\top \mathbf{P} \mathbf{x} + 2\left(\frac{1}{2}\mathbf{P}^{-1}\mathbf{q}\right)^\top \mathbf{P} \mathbf{x} + \mathbf{x}_0^\top \mathbf{P} \mathbf{x}_0 + r - \mathbf{x}_0^\top \mathbf{P} \mathbf{x}_0 = \tag{17}$$

$$= \mathbf{x}^\top \mathbf{P} \mathbf{x} + \mathbf{q}^\top \mathbf{x} + r. \tag{18}$$

Thus our original expression became:

$$(\mathbf{x}-\mathbf{x}_0)^\top \mathbf{P}(\mathbf{x}-\mathbf{x}_0) - d \leq 0 \tag{19}$$

We define $\mathbf{A} = \sqrt{\mathbf{P}}$ with SVD decomposition $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$.
Defining $\mathbf{z} = \mathbf{V}^\top(\mathbf{x}-\mathbf{x}_0)$ we get:

$$(\mathbf{x}-\mathbf{x}_0)^\top \mathbf{A}^\top \mathbf{A}(\mathbf{x}-\mathbf{x}_0) - d \leq 0 \tag{20}$$

$$(\mathbf{x}-\mathbf{x}_0)^\top \mathbf{V}\Sigma\mathbf{U}^\top \mathbf{U}\Sigma\mathbf{V}^\top(\mathbf{x}-\mathbf{x}_0) - d \leq 0 \tag{21}$$

$$(\mathbf{x}-\mathbf{x}_0)^\top \mathbf{V}\Sigma^2\mathbf{V}^\top(\mathbf{x}-\mathbf{x}_0) - d \leq 0 \tag{22}$$

$$\mathbf{z}^\top \Sigma^2 \mathbf{z} - d \leq 0 \tag{23}$$

$$\sum z_i^2 \sigma_i^2 \leq d \tag{24}$$

Defining $1/m_i^2 = \sigma_i^2/d$ we get:

$$\frac{z_1^2}{m_1^2} + \frac{z_2^2}{m_2^2} + ... + \frac{z_n^2}{m_n^2} \leq 1 \tag{25}$$

Homework

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

Further reading

- Symmetric Matrices and Eigendecomposition. Robert M. Freund, MIT, 2014.
- MOSEK, QP and QCQP.

Lecture slides are available via Github, links are on Moodle:

github.com/SergeiSa/Computational-Intelligence-2025