Notation

$\mathbb{R}$ - Set of real numbers
S^n - Set of symmetric $n \times n$ matrices
S^n_+ - Set of symmetric positive semidefinite $n \times n$ matrices
S^n_{++} - Set of symmetric positive definite $n \times n$ matrices
$M \succ 0$ - M is symmetric and positive definite
$M \succeq 0$ - M is symmetric and positive semidefinite
$C \bullet X$ - Matrix inner product $= Tr(C^T X)$

PSD - Positive Semidefinite

QCQP - Quadratically Constrained Quadratic Program
SDP - Semidefinite program
SOCP - Second Order Cone Program

WLOG - Without Loss Of Generality

Chapter 1

Introduction

In this thesis we will study quadratically constrained quadratic programs (QCQP)

Definition 1.1. The Quadratically Constrained Quadratic Program (QCQP) in the standard form is

$$\begin{array}{ll} \text{minimize} & x^T P_0 x + q_0^T x + r_0 \\ \text{subject to} & x^T P_k x + q_k^T x + r_k \leq 0, \ (k = 1, \ldots, m), \end{array} \tag{1.2}$$

where $x \in \mathbb{R}^n$ is a variable, and symmetric $n \times n$ matrices $P_0, P_1, \ldots, P_m \in S^n$, vectors $q_0, \ldots, q_m \in \mathbb{R}^n$ and scalars $r_1, \ldots, r_m \in \mathbb{R}$ are given.

The matrices $P_0, P_1, \ldots P_m$ are not necessarily positive semidefinite. Therefore the objective function as well as the constraints may be nonconvex. Generally, this problem is hard to solve. For example, the 0-1 constraint $x_i \in \{0, 1\}$ can be reformulated as $x^T e_i e_i^T x - e_i^T x = 0$. Thus QCQP includes 0-1 programming, which describes various NP-hard problems, such as knapsack, stable set, max cut etc. On the other hand, the above examples suggest that QCQP has many applications and is worth solving.

One of the possible approaches is relaxing QCQP to obtain problems which can be solved in polynomial time, namely linear programming (LP), second order cone programming (SOCP), or semidefinite programming (SDP).

1.1 Conic optimization classes

In this section we will introduce basic optimization classes mentioned above, in particular the linear programming (LP), second order cone programming (SOCP), or semidefinite programming (SDP). We will state the problems in standard forms and their duals. The dual problems will be only mentioned here and will be derived in the next section. We will also show that LP is subclass of convex QCQP, convex QCQP is subclass of SOCP, and SOCP is subclass of SDP, i.e.

$$LP \subseteq \text{convex } QCQP \subseteq SOCP \subseteq SDP. \tag{1.3}$$

1.1.1 Linear programming

When both, the objective and the constraint functions are linear (affine), the problem is called a linear program and it belongs to the Linear Programming class, or shortly LP. In this section we will introduce the standard form of LP and its dual. For reference and more information about this topic see i.e. [5] .

Definition 1.4. The primal–dual pair of linear programs in standard form is

$$\begin{array}{llcll} & \textit{Primal} & \qquad & & \textit{Dual} \\ \text{minimize} & c^T x, & & \text{maximize} & b^T y, \\ \text{subject to} & Ax = b, & & \text{subject to} & A^T y + s = c, \\ & x \in \mathbb{R}^n_+, & & & s \in \mathbb{R}^n_+, \end{array} \tag{1.5}$$

where $x \in R^n$, $y \in \mathbb{R}^m$, $s \in \mathbb{R}^n$ are the variables; the real $m \times n$ matrix A and vectors $b \in \mathbb{R}^m, c \in \mathbb{R}^n$ are given problem data.

Remark 1.6. Linear programs can be formulated in various forms (including $\geq, \leq$ inequalities, free variables, possibly some linear fractions in objective) but all of them can be transformed to the standard form.

Relation to previous classes

Linear programming is a special case of QCQP, when all matrices in quadratic forms are 0.

1.1.2 Second order cone programing

The second order cone programming (SOCP) is a convex optimization class which can be solved with great efficiency using interior point methods. In this section we will introduce the standard form of SOCP and its dual. For reference and more information about this topic see [5] .

Let us first define second order cone.

Definition 1.7 (Second order cone)**.** We say Q_n is second order cone of dimension n if

$$Q_n = \{x \in \mathbb{R}^n \mid x = (x_0, \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1}, \|\bar{x}\|_2 \leq x_0\}. \tag{1.8}$$

Definition 1.9 (SOCP)**.** The primal–dual pair of the Second Order Cone Program (SOCP) in the standard form is

$$
\begin{array}{llcll}
 & \textit{Primal} & & & \textit{Dual} \\
\text{minimize} & c^T x, & & \text{maximize} & b^T y, \\
\text{subject to} & Ax = b, & & \text{subject to} & A^T y + s = c, \\
 & x \in Q, & & & s \in Q,
\end{array}
\tag{1.10}
$$

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ and $s \in \mathbb{R}^n$ are the variables; and $m \times n$ real matrix A, vectors $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and second order cone Q are given problem data.

Remark 1.11. Second order cone programs can be formulated in various forms (including quadratic objective or several second order cone constraints of the affine functions), but all of them can be transformed to the standard form.

Remark 1.12. In general, any program of the form

$$
\begin{array}{llcll}
 & \textit{Primal} & & & \textit{Dual} \\
\min & c^{1T} x^1 + \cdots + c^{kT} x^k, & & \max & b^T y, \\
\text{s.t.} & A^1 x^1 + \cdots + A^k x^k = b, & & \text{s.t.} & A^{iT} y + s^i = c^i, \\
 & x^i \in Q_{n_i},\ (i = 1, \ldots, k), & & & s^i \in Q_{n_i},\ (i = 1, \ldots, k),
\end{array}
\tag{1.13}
$$

is considered to be SOCP. The second order cone constraints can be also formulated as, $x = (x^1, \ldots, x^k) \in Q$, where Q is Cartesian product of second order cones,

$$Q = Q_{n_1} \times Q_{n_2} \times \cdots \times Q_{n_k}, \tag{1.14}$$

Since, such Q has all important properties of second order cone (see appendix), and algorithmic aspects of solving standard SOCP also work for this more general case [7,8]. In order to keep things simple, we will sometimes consider only the standard form stated in the Definition 1.9, but all the details can be also done for this more general form.

Relation to previous classes

Second Order Cone Programming includes convex LP as special case. We will show that SOCP in fact includes convex QP as a subclass. We will demonstrate procedure proposed in [4] used to reformulate convex QP as SOCP.

Let us have convex QCQP. In other words, suppose that $n \times n$ matrices A_k, $k = 0, \ldots, m$ are positive semidefinite.

$$
\begin{array}{ll}
\text{minimize} & x^T P_0 x + q_0^T x + r_0, \\
\text{subject to} & x^T P_k x + q_k^T x + r_k \leq 0,\ (k = 1, \ldots, m),
\end{array}
\tag{1.15}
$$

First of all, rewrite problem equivalently as

$$\begin{array}{ll} \text{minimize} & t, \\ \text{subject to} & x^T P_0 x + q_0^T x + r_0 \le t, \\ & x^T P_k x + q_k^T x + r_k \le 0, \ (k = 1, \ldots, m). \end{array} \tag{1.16}$$

To avoid tedious notation, without loss of generality, suppose that considered program already has linear objective function (i.e. $P_0 = 0$). Also suppose that we have separated all the linear constraints (ones where $P_k = 0$) and arrange them into more compact form $Ax = b$. Even if we did not, the following procedure will still be correct, but will result in more complicated formulation of linear constraints.

Each convex quadratic constraint

$$x^T P x + q^T x + r \le 0 \tag{1.17}$$

can be transformed into the second order cone constraint. Suppose that $P \neq 0$ and rank $P = h$. Then there exists $n \times h$ matrix L such that $P = LL^T$. Such L can be computed by Choelsky factorization of P. Now rewrite (1.17) as

$$(L^T x)^T (L^T x) \le -q^T x - r. \tag{1.18}$$

It can be easily verified that $w \in \mathbb{R}^t$, $\xi \in \mathbb{R}$ and $\eta \in \mathbb{R}$ satisfy

$$w^T w \le \xi\eta, \quad \xi \ge 0 \quad \text{and} \quad \eta \ge 0$$

if and only if they satisfy

$$\left\| \begin{pmatrix} \xi - \eta \\ 2w \end{pmatrix} \right\|_2 \le \xi + \eta.$$

If we take $w = L^T x$, $\xi = 1$ and $\eta = -q^T x - r$, then inequality (1.18) is equivalent to the second order cone constraint

$$\|v\|_2 \le v_0, \quad \text{where} \quad \begin{pmatrix} v_0 \\ v \end{pmatrix} = \begin{pmatrix} 1 - q^T x - r \\ 1 + q^T x + r \\ 2L^T x \end{pmatrix} \in \mathbb{R}^{h+2}. \tag{1.19}$$

Now the intersection of all such second order cone constraints can be easily expressed as Cartesian product of second order cones, thus we have obtained problem of SOCP in form 1.13.

1.1.3 Semidefinite programming

The semidefinite programming (SDP) is a convex optimization class which can be solved efficiently using interior point methods. In this section we will introduce the standard form of SDP and its dual. For reference and more information about this topic see [5] .

Firstly, let us introduce notation we will use to simplify the standard form.

Definition 1.20. Let A, X be real $n \times m$ matrices, we will denote their inner product

$$A \bullet X = Tr(A^T X).$$

Where $Tr(M)$ denotes trace of matrix M i.e. sum of the diagonal elements of M.

Definition 1.21 (SDP)**.** The primal–dual pair of the Semidefinite Program (SDP) in the standard form is

$$\begin{array}{llcll} & \textit{Primal} & & & \textit{Dual} \\ \text{minimize} & A_0 \bullet X, & & \text{maximize} & b^T y, \\ \text{subject to} & A_k \bullet X = b_k, & & \text{subject to} & \sum_{k=1}^{m} y_k A_k + S = A_0, \\ & (\text{for } k = 1, \ldots, m), & & & S \in \mathbb{S}_+^n, \\ & X \in \mathbb{S}_+^n, & & & \end{array} \tag{1.22}$$

where $X \in \mathbb{S}_+^n$, $y = (y_1, \ldots, y_m)^T \in \mathbb{R}^m$ and $S \in \mathbb{S}^n$ are the variables; and symmetric matrices $A_0, A_1, \ldots, A_m \in \mathbb{S}^n$ and scalars $b_1, \ldots, b_m \in \mathbb{R}$ are given.

Surprisingly, the variable in SDP is a symmetric matrix (not a vector). In order to be consistent with other classes we will sometimes use a *svec* operator.

Definition 1.23. We define operator $svec : \mathbb{S}^n \to \mathbb{R}^{n(n+1)/2}$, such that for any $n \times n$ symmetric matrix M

$$svec(M) = (\delta_{11} M_{11}, \delta_{12} M_{12}, \delta_{22} M_{22}, \ldots, \delta_{1n} M_{1n}, \ldots, \delta_{nn} M_{nn})^T, \tag{1.24}$$

where

$$\delta_{ij} = \begin{cases} 1, & \text{if } \ i = j, \\ \sqrt{2}/2, & \text{otherwise.} \end{cases} \tag{1.25}$$

Notice that δ_{ij} are defined cleverly, so that inner product of symmetric matrices is equivalent to the standard inner product of their images

$$A \bullet X = svec(A)^T svec(X), \tag{1.26}$$

for any pair of symmetric matrices $A, X \in \mathbb{S}^n$. Now we can easily formulate the problems of SDP in terms of standard inner product over the space of real vectors.

$$\begin{array}{ll} \text{minimize} & svec(A_0)^T svec(X), \\ \text{subject to} & svec(A_k)^T svec(X) = b_k, \ (k = 1, \ldots, m), \\ & svec(X) \in \mathcal{K}(\mathbb{S}_+^n), \end{array} \tag{1.27}$$

where $svec(X)$ is variable, $\mathcal{K}(\mathbb{S}_+^n) = \{svec(U) \in \mathbb{R}^{n(n+1)/2} \mid U \in \mathbb{S}_+^n\}$ and problem data are from the standard SDP (1.22).

Relation to previous classes

The SDP primal-dual pair looks suspiciously similar to the both LP and SOCP primal-dual pairs. The only difference between LP and SOCP is the nonnegative orthant is replaced by second order cone. The SDP, in the *svec* form (1.27), further generalizes the cone constraint with the semidefinite cone.

In fact, SOCP is subclass of SDP. We will show how the standard SOCP can be rewritten as SDP. First of all, instead of minimizing $c^T x$ we will minimize t with additional constraint $t \geq c^T x$.

The only nontrivial part is to rewrite conic constraint

$$x \in Q_n \;\Leftrightarrow\; ||\bar{x}|| \leq x_1 \;\Leftrightarrow \left\{ \begin{array}{c} \bar{x}^T \bar{x} \leq x_1^2 \\ 0 \leq x_1 \end{array} \right\} \tag{1.28}$$

$$\Leftrightarrow \left\{ \begin{array}{r} \frac{\bar{x}^T \bar{x}}{x_1} \leq x_1 \\ 0 \leq x_1 \end{array} \right\} \Leftrightarrow \begin{pmatrix} x_1 & \bar{x}^T \\ \bar{x} & x_1 I_{n-1} \end{pmatrix} \succeq 0. \tag{1.29}$$

Where last equivalence is provided by Schur complement lemma (see appendix, Theorem A.1). From here, it can be easy brought to the standard form (see appendix).

In case of more general standard form of SOCP (1.13), the $x \in Q$ constraint can be transformed similarly.

$$x = (x^1, \ldots, x^k)^T \Leftrightarrow M = diag(M_1, \ldots, M_k) \succeq 0, \tag{1.30}$$

where M is a block diagonal matrix, with blocks M_i of the form (1.29), for $i = 1, \ldots, k$, corresponding to the constraints $x^i \in Q_{n_i}$.

1.1.4 Conic programming

The previous mentioned classes are quite similar. With respect to their variable space, all of them have linear objective, linear constraints and cone constraint.

In fact, they are special cases of the so called conic linear programs.

Definition 1.31 (Conic Programming)**.** The primal–dual pair of the Linear Conic Program in the standard form is

$$\begin{array}{llcll} & \textit{Primal} & \qquad & & \textit{Dual} \\ \text{minimize} & c^T x, & & \text{maximize} & b^T y, \\ \text{subject to} & Ax = b, & & \text{subject to} & A^T y + s = c, \\ & x \in \mathcal{K}, & & & s \in \mathcal{K}^*, \end{array} \tag{Conic Program}$$

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ and $s \in \mathbb{R}^n$ are the variables; and $m \times n$ real matrix A, vectors $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and the proper cone $\mathcal{K}$ are given problem data.

The

$$K^* = \{z \mid \forall x \in \mathcal{K},\ x^T z \geq 0\}, \tag{1.32}$$

denotes the dual cone of $\mathcal{K}$ (see appendix A.2).

Conic programming contains, but is not limited to, any problems combined from LP, SOCP and SDP programs. For example

$$\begin{array}{ll} \text{minimize} & c^T x, \\ \text{subject to} & A^i x^i = b^i,\ (i = 1, \dots, k), \\ & x = (x^1, \dots, x^k) \in \mathcal{K} = (\mathcal{K}^1, \dots, \mathcal{K}^k), \end{array} \tag{1.33}$$

where the variable $x = (x^1, \dots, x^k)^T$ is the Cartesian product of the variables x^i, constrained by various LP, SOCP or SDP constraints $A^i x^i = b$, $x_i \in \mathcal{K}^i$, where each $\mathcal{K}^i$ is either nonegative orthant, second order cone or semidefinite cone.

This is due to the fact, that all cones we have talked about so far are proper cones (nonegative orthant $\mathbb{R}^n_+$, second order cone Q_n and semidefinite cone $\mathbb{S}^n_+$ as a subset of $\mathbb{R}^{n(n+1)/2}$). For proper cones it holds that their Cartesian product is again proper cone (see Section A.2 in appendix).

1.2 Duality

1.2.1 Duality in conic programming

We will derive the dual forms of LP, SOCP, SDP all at once by deriving Lagrange dual of general conic program

$$\begin{array}{ll} \text{minimize} & c^T x, \\ \text{subject to} & Ax = b, \\ & x \in \mathcal{K}. \end{array} \tag{1.34}$$

The Lagrangian of the problem is given by $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathcal{K}^* \to \mathbb{R}$,

$$\mathcal{L}(x, y, s) = c^T x + y^T (b - Ax) - s^T x. \tag{1.35}$$

The last term (notice that $s \in \mathcal{K}^*$) is added to take account of the conic constraint $x \in \mathcal{K}$. It is with negative sign in order to have $\mathcal{L}(x, \cdot, \cdot) \leq c^T x$ for all x feasible in (1.34). Indeed, from the very definition of dual cone:

$$\sup_{s \in \mathcal{K}^*} -s^T x = \begin{cases} 0 & \text{if } x \in \mathcal{K}, \\ +\infty & \text{otherwise.} \end{cases} \tag{1.36}$$

Therefore, the Lagrange dual function is

$$\begin{aligned} g(y,s) &= \inf_x \ \mathcal{L}(x,y,s) && (1.37)\\ &= \inf_x \ y^T b + (c + A^T y - s)^T x && (1.38)\\ &= \begin{cases} b^T y & \text{if } c - A^T y - s = 0,\\ -\infty & \text{otherwise.}\end{cases} && (1.39)\end{aligned}$$

Hence, the dual problem of linear conic programming in the standard form is

$$\begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to} & A^T y + s = c \\ & s \in \mathcal{K}^* \end{array} \qquad (1.40)$$

Since $\mathbb{R}^n_+$, Q_n and $\mathbb{S}^n_+$ are self-dual, by replacing the $\mathcal{K}$ (and $\mathcal{K}^*$) with any of these cones, we get the dual of standard LP, SOCP and SDP as given in the Section 1.1.

1.2.2 Duality in QCQP

We will derive dual form of standard QCQP

$$\begin{array}{ll} \text{minimize} & x^T P_0 x + q_0^T x + r_0 \\ \text{subject to} & x^T P_k x + q_k^T x + r_k \le 0, \ (k = 1, \dots, m) \end{array} \qquad (1.41)$$

The Lagrangian of the problem is given by $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m_+ \to \mathbb{R}$,

$$\begin{aligned} \mathcal{L}(x,y) &= x^T P_0 x + q_0^T x + r_0 + \sum_{k=1}^m y_k (x^T P_k x + q_k^T x + r_k) && (1.42)\\ &= x^T P(y) x + q(y)^T x + r(y), && (1.43)\end{aligned}$$

where

$$P(y) = P_0 + \sum_{k=1}^m y_k P_k, \quad q(y) = q_0 + \sum_{k=1}^m y_k q_k, \quad r(y) = r_0 + \sum_{k=1}^m y_k r_k. \qquad (1.44)$$

It holds that $\inf_x \mathcal{L}(x,y) > -\infty$ if and only if $P(y) \succeq 0$ and there exists $\hat{x}$ such that $P(y)\hat{x} + q(y) = 0$.

Thus, the Lagrange dual function is

$$\begin{aligned} g(y) &= \min_x \ \mathcal{L}(x,y) && (1.45)\\ &= \begin{cases} -\frac{1}{4} q(y)^T P(y)^\dagger q(y) + r(y) & \text{if } P(y) \succeq 0, \ \ q(y) \in \mathcal{R}(P(y))\\ -\infty & \text{otherwise,}\end{cases} && (1.46)\end{aligned}$$

where $P^\dagger$ denotes Moore-Penrose pseudoinverse of P (see appendix). Finally, dual form of standard QCQP problem is

$$\begin{array}{ll} \text{maximize} & -\frac{1}{4}q(y)^T P(y)^\dagger q(y) + r(y), \\ \text{subject to} & y \geq 0, \\ & P(y) \succeq 0, \\ & \mathcal{R}(q(y)) \subseteq \mathcal{R}(P(y)), \end{array} \tag{QCQP Dual}$$

where $y \in R^m$ is variable; and problem data $P_0, P_1 \ldots, P_m, q_0, q_1 \ldots, q_m, r_0, r_1 \ldots, r_m$ are given from the primal QCQP above.

This dual problem is basically a SDP (in the LMI form). We first rewrite the objective as linear function t with additional constraint.

$$\begin{array}{ll} \text{maximize} & t, \\ \text{subject to} & t \leq -\frac{1}{4}q(y)^T P(y)^\dagger q(y) + r(y), \\ & y \geq 0, \\ & P(y) \succeq 0, \\ & \mathcal{R}(q(y)) \subseteq \mathcal{R}(P(y)). \end{array} \tag{1.47}$$

Due to the Schur complement lemma (see appendix, Theorem A.1) the above is equivalent to

$$\begin{array}{ll} \text{maximize} & t, \\ \text{subject to} & M := \begin{pmatrix} r(y) - t & \frac{1}{2}q(y)^T \\ \frac{1}{2}q(y) & P(y) \end{pmatrix} \succeq 0, \\ & y \geq 0, \end{array} \tag{1.48}$$

where the matrix M is easily expanded as

$$M = M(t) + \sum_{k=1}^{n} M(y_k) \tag{1.49}$$

with

$$M(t) = \begin{pmatrix} -t & 0 \\ 0 & 0 \end{pmatrix}, \quad M(y_k) = \begin{pmatrix} r(y_k) & \frac{1}{2}q(y_k)^T \\ \frac{1}{2}q(y_k) & P(y_k) \end{pmatrix}. \tag{1.50}$$

1.3 Relaxations

As mentioned in the beginning, our strategy is to relax the nonconvex QCQPs (1.2) to easier problem. Let us first explore what is relaxation and how can it be useful.

Relaxation is usually freely understood as an optimization problem which is obtained by relaxing (loosening) some constraints or even by approximating objective function with a different one. The goal is to obtain a problem which is easier to solve, but still carries some kind of information about the original problem. For example, solving the relaxation may give an approximation of the original problem solution.

One could say, that relaxation of minimization problem

$$\begin{array}{ll} \text{minimize} & f(x), \\ \text{subject to} & x \in X, \end{array} \tag{1.51}$$

is another minimization problem

$$\begin{array}{ll} \text{minimize} & f_R(x), \\ \text{subject to} & x \in X_R, \end{array} \tag{1.52}$$

with properties $X \subseteq X_R$ and $c_R(x) \leq c(x) \forall x \in X$. It easily follows, that solving the relaxed problem will provide a lower bound on the optimal value of original problem. In some cases we can also extract a feasible solution of the original problem from solution of the relaxation. In that case we obtain an upper bound for the optimal value.

Moreover, these bounds may not only give us an idea about the optimal value, but also, may provide means to find an optimal solution of the original problem.

1.4 SDP relaxation of QCQP

Consider QCQP (1.2). Using identity $x^T P_k x = P_k \bullet xx^T$, which follows from the Definiton 1.20, it can be rewritten as follows

$$\begin{array}{ll} \text{minimize} & P_0 \bullet X + q_0^T x + r_0, \\ \text{subject to} & P_k \bullet X + q_k^T x + r_k \leq 0, \ (k = 1, \ldots, m) \\ & X = xx^T. \end{array} \tag{1.53}$$

Notice, that variable X is symmetric $n \times n$ matrix. The problem can be reformulated in the following way:

$$\begin{array}{ll} \text{minimize} & M_0 \bullet Y, \\ \text{subject to} & M_k \bullet Y \leq 0, \ (k = 1, \ldots, m), \\ & X = xx^T, \end{array} \tag{1.54}$$

where ,

$$M_k = \begin{pmatrix} \alpha_k & \frac{1}{2} q_k^T \\ \frac{1}{2} q_k & P_k \end{pmatrix}, \qquad Y = \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix}. \tag{1.55}$$

This problem still has a non-convex constraint $X = xx^T$, which can be relaxed by a convex constraint, as stated in the following lemma.

Lemma 1.56. *Let $x \in \mathbb{R}^n$, an $n \times n$ symmetric matrix X, and $n+1 \times n+1$ symmetric matrix Y such that*

$$Y = \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix}.$$

Then

(i) $X \succeq xx^T$ *if and only if* $Y \succeq 0$.

(ii) $X = xx^T$ *holds if and only if* $Y \succeq 0$ *and* $\operatorname{rank} Y = 1$.

Proof. (i) The statement follows from Schur complement lemma for PSD (see appendix, Theorem A.1 in appendices)

(ii) ($\Rightarrow$) If $X = xx^T$, then also $X \succeq xx^T$, thus $Y \succeq 0$ holds by (i). And since $X = xx^T$,

$$Y = \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix} = \begin{pmatrix} 1 \\ x \end{pmatrix} (1,\ x^T).$$

Hence $\operatorname{rank} Y = 1$.
($\Leftarrow$) Let $Y \succeq 0$ and $\operatorname{rank} Y = 1$. Since $\operatorname{rank} Y = 1$, each row of Y must be scalar multiple of the first (obviously nonzero) row $(1,\ x^T)$. To match the first column the $(i+1)$-st row of Y must be $x_i(1,\ x^T)$, for $i = 1, \dots, n$. Therefore, $X = xx^T$. □

Remark 1.57. Notice that we have proven last implication of (ii) without using $Y \succeq 0$. In fact it is redundant. It also holds that $X = xx^T \Leftrightarrow \operatorname{rank} Y = 1$. In fact, there are only 2 options for Y of rank 1: $Y = vv^T$ or $Y = -vv^T$. The second option is easily excluded, because $Y_{11} = 1 > 0$. However, this redundant constraint $Y \succeq 0 \Leftrightarrow X \succeq xx^T$ will let us keep something from the rank 1 constraint after relaxing it. This approach of adding the redundant constraints (also known as valid inequalities) is often usefull for strengthening the relaxation. For more about valid inequalities see [12, 13, 14].

Using the Lemma 1.56 and relaxing the nonconvex rank 1 constrain we get the following SDP relaxation of (1.2)

$$\begin{array}{ll} \text{minimize} & M_0 \bullet Y \\ \text{subject to} & M_k \bullet Y \le 0,\ (k = 1, \dots, m) \\ & Y \succeq 0. \end{array} \tag{1.58}$$

Expanding the terms $M_k \bullet Y$ we obtain

$$\begin{array}{ll} \text{minimize} & P_0 \bullet X + q_0^T x + r_0 \\ \text{subject to} & P_k \bullet X + q_k^T x + r_k \le 0,\ (k = 1, \dots, m) \\ & X \succeq xx^T. \end{array} \tag{1.59}$$

The above relaxed problem has different variable space $(X, x) \in \mathbb{R}^{n \times n} \times \mathbb{R}^n$ than original problem $x \in \mathbb{R}^n$. In other words, the variable space increased from $O(n)$ to $O(n^2)$ variables.

Semidefinite program relaxations can provide tight bounds, but they can also be expensive to solve by classical interior point methods (because of the $O(n^2)$ variables). Many researchers have suggested various alternatives to interior point methods for improving solution times. Others have studied different types of relaxations, for example, ones based on LP or SOCP. [1]

1.5 SOCP relaxation

These relaxations are expected to provide weaker bounds in less time compared to SDP relaxations. In fact, SOCP relaxations are often constructed as further relaxations of SDP relaxations. So, in certain sense, one can see that SOCP relaxations are never tighter than their SDP counterparts. [1]

Using the procedure from section 1.1.2, any convex QCQP relaxation can be easily formulated as SOCP relaxation. Therefore we can consider any convex QCQP relaxation as SOCP relaxation. Specifically, such a convex QCQP relaxation may be represented as

$$\begin{array}{ll} \text{minimize} & x^T B_0 x + b_0^T x \\ \text{subject to} & x^T B_k x + b_k^T x + \beta_i \leq 0, \ (k = 1, \dots, l) \end{array} \tag{1.60}$$

where all $B_k \succeq 0$ for $k = 0, \dots, l$. We say that (1.60) is SOCP relaxation of (1.2) if that x is feasible for (1.2) implies that x is feasible for (1.60) and $x^T B_0 x + b_0^T x \leq x^T A_0 x + a_0^T x$ holds.

It is well known that SOCPs can also be solved as SDP. In [4] it is shown, that (1.60) is equivalent to the SDP

$$\begin{array}{ll} \text{minimize} & B_0 \bullet X + b_0^T x \\ \text{subject to} & B_k \bullet X + b_k^T x + \beta_i \leq 0, \ (k = 1, \dots, l) \\ & X \succeq x x^T. \end{array} \tag{1.61}$$

In [4] authors provided SOCP relaxation of QCQP (1.2) in original variable space $x \in \mathbb{R}^n$. First they WLOG assume that objective function is linear (otherwise we can introduce new variable $t \geq x^T A_0 x + a_0^T$ and then minimize t). Then each A_k is written as

$$A_k = A_k^+ - A_k^-, \quad \text{where} \quad A_k^+, A_k^- \succeq 0, \quad k = 1, \dots, m.$$

So that each constraint can be expressed as

$$x^T A_k^+ x + a_k^T x + \alpha_k \leq x^T A_k^- x.$$

Then an auxiliary variable $z_k \in \mathbb{R}$ is introduced to represent $x^T A_k^- x = z_k$, but also immediately relaxed as $x^T A_k^- x \leq z_k$, resulting in covnex system

$$\begin{array}{rl} x^T A_k^+ x + a_k^T x + \alpha_k & \leq z_k \\ x^T A_k^- x & \leq z_k. \end{array} \tag{1.62}$$

Finally, z_k must be bounded in some fashion, say as $z_k \leq \mu \in \mathbb{R}$, or else the relaxation would be useless. In this way convex QCQP relaxation is constructed and it is simply transformed to SOCP using procedure from [4] described in section 1.1.2.

Remark 1.63. Technically it is convex QCQP relaxation, but it is usually solved in SOCP form by algorithms for SOCP, hence it is referred to as SOCP relaxation.

1.5.1 SOCP relaxation of semidefinite constraint

Semidefinite constraint $X - xx^T \succeq 0$ in (1.59) is equivalent to $C \bullet (X - xx^T) \geq 0$ for all $C \in S_+^n$. Using this fact, authors of [4] propose SOCP relaxation of the semidefinite constraint $X - xx^T \succeq 0$ by replacing it with multiple constraints of the form

$$x^T C_i x - C \bullet X \leq 0 \quad (i = 1, \dots, l).$$

Since $C_i \succeq 0$, these are convex quadratic constraints and using the procedure described earlier (in section 1.12) one can formulate them as second order cone constraints of the form

$$\begin{pmatrix} v_0^i \\ v^i \end{pmatrix} = \begin{pmatrix} 1 + C_i \bullet X \\ 1 - C_i \bullet X \\ 2L_i^T x \end{pmatrix}, \quad ||v^i|| \leq v_0^i, \quad (i = 1, \dots, l) \tag{1.64}$$

For further details about strength of this approach and suggested choice of matrices C_i based on problem data see [4].

1.6 Mixed SOCP-SDP relaxation

In [1] authors have introduced compromise, relaxation of QCQP (1.2) somewhere between SDP and SOCP. We will describe their approach on specific case they provide as gentle introduction. We are dealing with (1.2)

$$\begin{array}{ll} \text{minimize} & x^T A_0 x + a_0^T x \\ \text{subject to} & x^T A_k x + a_k^T x + \alpha_k \leq 0, \ (k = 1, \dots, m) \end{array} \tag{1.65}$$

Let $y_{min}(A_k)$ denote smallest eigenvalue of A_k. For all $k = 0, \dots, m$ define $y_k = -y_{min}(A_k)$ so that $A_k + I y_k \succeq 0$. Then (1.2) is equivalent to

$$\begin{array}{ll} \text{minimize} & -y_0 x^T x + x^T (A_0 + y_0 I) x + a_0^T x \\ \text{subject to} & -y_k x^T x + x^T (A_k + y_k I) x + a_k^T x + \alpha_k \leq 0, \ (k = 1, \dots, m) \end{array}$$

which has following SOCP-SDP relaxation

$$\begin{array}{ll} \text{minimize} & -y_0 Tr(X) + x^T(A_0 + y_0 I)x + a_0^T x \\ \text{subject to} & -y_k Tr(X) + x^T(A_k + y_k I)x + a_k^T x + \alpha_k \leq 0, \\ & (k = 1, \dots, m) \\ & X \succeq xx^T \end{array} \tag{1.66}$$

Notice that other than $X \succeq xx^T$, the only variables in X to appear in the program are diagonal elements X_{jj}. Also, one can see that with fixed x the diagonal entries of X can be made arbitrarily large to satisfy all constraints with $y_k > 0$, as well as drive objective to $-\inf$ if $y_0 > 0$. Therefore, in general, X_{jj} should be bounded to form a sensible relaxation. In the paper they use $x_j \in [0, 1]$ and $X_j j \leq x_j$ to establish boundedness.

Next proposition from [11] gives equivalent formulation of $X \succeq xx^T$ constraint, only in terms of x and diagonal entries of X.

Proposition 1.67. *Given a vector x and scalars $X_{11}, \dots, X_{nn}$, there exists a symmetric-matrix completion $X \in S^n$ of $X_{11}, \dots, X_{nn}$ satisfying $X \succeq xx^T$ if and only if $X_{jj} \geq x_j^2$ for all $j = 1, \dots, n$.* [11]

Thus, in light of this proposition, the problem with additional bounding constraints $X_{jj} \leq x_j$, the problem (1.66) is equivalent to

$$\begin{array}{ll} \text{minimize} & -y_0 Tr(X) + x^T(A_0 + y_0 I)x + a_0^T x \\ \text{subject to} & -y_k Tr(X) + x^T(A_k + y_k I)x + a_k^T x + \alpha_k \leq 0, \\ & (k = 1, \dots, m) \\ & x_j^2 \leq X_{jj} \leq x_j \quad (j = 1, \dots, n) \end{array} \tag{1.68}$$

Compared to SDP relaxation (1.59), which has $O(n^2)$ variables, problem (1.68) has only $O(n)$ and hence is much faster to solve. On the other hand bound should be generally weaker than the SDP bound.

In the paper [1] is this approach further generalized and explored. Splitting $A = -D + (A + D)$ (instead of $-y_k I + (A + y_k I)$) with clever choice of C-block diagonal matrix D.

1.7 Using relaxations to solve the original problem

1.7.1 Branch and bound

The basic idea of this method is to divide and conquer. There are 3 basics steps we will do repeatedly. Branch, compute lower bound, compute upper bound and prune.

- Branch. First divide by branching original problem into smaller subproblems i.e. by partition of the feasible set. One can do this for example by adding box constraints $l_i^j \leq x_i \leq u_i^j$, obtaining a new subproblem for each $j \in J$. It is important that union of feasible sets of all subproblems covers whole feasible set of original problem.

- Compute lower bound. Compute lower bounds for each subproblem (i.e. by solving relaxed problem).

- Compute upper bound and prune. Compute some upper bound for the optimal value (i.e. compute feasible solution for chosen subproblem or try to extract it from solution of relaxed problem). Having both lower and upper bounds, we can conquer by pruning all the branches with lower bound greater then this upper bound. The optimal solution is surely not in these branches.

- Repeat. This process is repeated by branching remaining subproblems, computing new bounds for smaller problems, improving bounds and further pruning the tree of problems.

One can use more or less clever techniques to decide when, how and which subproblems to branch or how and when to compute bounds.

Appendix A

A.1 Schur complement

Theorem A.1 (Schur complement lemma for PSD)**.** *Let M be a symmetric matrix of the form*

$$M = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}.$$

The following conditions are equivalent:

1. $M \succeq 0$ *(M is positive semidefinite).*
2. $A \succeq 0, \quad (I - AA^\dagger)B = 0, \quad C - B^T A^\dagger B \geq 0.$
3. $A \succeq 0, \quad \mathcal{R}(B) \subseteq \mathcal{R}(A), \quad C - B^T A^\dagger B \geq 0.$

(The roles of A and C can be switched.) Where $A^\dagger$ denotes pseudoinverse of A and $\mathcal{R}(B)$ denotes column range of B. [10, 5]

Proof. (will be added later)

A.2 Cones

Definition A.2 (Cone)**.** Set $\mathcal{K}$ is cone if for all $x \in \mathcal{K}$ and all $\theta \geq 0$, it holds that $\theta x \in \mathcal{K}$.

Definition A.3 (Proper cone)**.** The cone $\mathcal{K} \subseteq \mathbb{R}^n$ is a proper cone if it has following properties

- $\mathcal{K}$ is closed
- $\mathcal{K}$ is convex (for any $\theta_1, \theta_2 \geq 0$ and $x_1, x_2 \in \mathcal{K}$ also $\theta_1 x_1 + \theta_2 x_2 \in \mathcal{K}$)

- $\mathcal{K}$ has nonempty interior ($int\mathcal{K} \neq \emptyset$)
- $\mathcal{K}$ is pointed (does not contain whole line, i.e. if $\pm x \in \mathcal{K}$, then $x = 0$).

Definition A.4 (Dual cone)**.** For any cone $\mathcal{K}$ we define its dual cone $\mathcal{K}^*$ as a set

$$\mathcal{K}^* = \{z \mid \forall x \in \mathcal{K},\ x^T z \geq 0\}. \tag{A.5}$$

If $\mathcal{K} = \mathcal{K}^*$ we say cone $\mathcal{K}$ is selfdual.

TODO:

- carthesian product of proper/self-dual cones
- $\mathbb{R}^n_+$, Q_n, $\mathbb{S}^n_+$ are proper, self-dual cones + figures.
- example of converting SOCP to SDP
- example of converting QP to SOCP
- Moore-Penrose pseudoinverse

References

[1] S. Burer, S.Kim, M. Kojima, *Faster, but Weaker, Relaxations for Quadratically Constrained Quadratic Programs,* April, 2013
http://www.is.titech.ac.jp/~kojima/articles/socpsdp.pdf

[2] S. Kim, M. Kojima, *Exact Solutions of Some Nonconvex Quadratic Optimization Problems via SDP and SOCP Relaxations,*January 2002.Computational Optimization and Applications Vol.26 (2) 143-154 (2003)
http://www.is.titech.ac.jp/research/research-report/B/B-375.ps.gz

[3] S. Kim, M. Kojima, M. Yamashita, *Second Order Cone Programming Relaxation of a Positive Semidefinite Constraint,* July 2002. Optimization Methods and Software Vol.18 (5) 535-541 (2003).
http://www.is.titech.ac.jp/~kojima/articles/B-381.pdf

[4] S. Kim, M. Kojima, *Second Order Cone Programming Relaxations of Nonconvex Quadratic Optimization Problems,* Optim. Methods Softw., 15(3-4):201-224, 2001
http://www.researchgate.net/publication/2637515_Second_Order_Cone_Programming_Relaxation_of_Nonconvex_Quadratic_Optimization_Problems

[5] S. Boyd, L. Vandenberghe, *Convex Optimization,* Cambridge University Press 2004, seventh printing with corrections 2009
http://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf

[6] R. M. Freund, *Introduction to Semidefinite Programming (SDP),* 2009
http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-251j-introduction-to-mathematical-programming-fall-2009/readings/MIT6_251JF09_SDP.pdf

[7] F. Alizadeh, D. Goldfarb, *Second-order cone programming,* Springer, 2002

[8] K. Pokorná, *Second order cone programing, (Master's thesis),* 2013
`http://www.iam.fmph.uniba.sk/studium/efm/diplomovky/2013/pokorna/diplomovka.pdf`

[9] M. S. Lobo, L. Vandenberghe, S. Boyd, H. Lebret, *Applications of second-Order cone programming,* Linear Algebra and its Applications 284, 1998, 193-228
`http://www.seas.ucla.edu/~vandenbe/publications/socp.pdf`

[10] J. Gallier, *The Schur Complement and Symmetric Positive Semidefinite (and Definite) Matrices,* December 10, 2010
`http://www.cis.upenn.edu/~jean/schur-comp.pdf`

[11] R. Grone, C. Johnson, E. Sá, H. Wolkowicz, *Positive definite completions of partial hermitian matrices,* Linear Algebra Applications, 58:109-124, 1984.

[12] H. Wolkowicz, R. Saigal,L. Vandenberghe, *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications,* ISBN 978-1-4615-4381-7, second printing 2003

[13] B. Kocuk, S. S. Dey, X. A. Sun, *Inexactness of SDP Relaxation and Valid Inequalities for Optimal Power Flow,* IEEE Transactions on Power Systems, 2014

[14] H. Dong, J. Linderoth, *On Valid Inequalities for Quadratic Programming with Continuous Variables and Binary Indicators* Springer Volume 7801 of the series Lecture Notes in Computer Science pp 169-180,