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 (QCQP)**.** We say minimization problem is Quadratically Constrained Quadratic Program (or shortly just QCQP) if it has the form of

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

Where $x \in \mathbb{R}^n$ is 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.

Matrices $P_0, P_1, \ldots P_m$ are not necessarily positive semidefinite. Therefore the objective function as well as constraints may be nonconvex. Generally this problem is hard to solve. For example, 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, this suggests that QCQP has many applications and is worth solving.

To this end, often used approach 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 problems (LP, SOCP, SDP) in standard forms and their duals. However, duals will be only mentioned here, we will derive them in the next section. We will also note their relations to QCQP and each other.

1.1.1 Linear programming

When both objective and constraint functions are linear (affine), the problem is called linear program or shortly LP. In this section we will introduce standard and dual form of LP, and few important properties. For reference and more information about this topic please see [5, 6,] .

Definition 1.2. We say minimization problem is Linear program (or shortly LP) if it has form of

$$
\begin{array}{ll|ll}
\multicolumn{2}{c|}{\textit{Primal}} & \multicolumn{2}{c}{\textit{Dual}} \\
\text{minimize} & c^T x & \text{maximize} & b^T y \\
\text{subject to} & Ax = b, & \text{subject to} & A^T y + s = c, \\
& x \in \mathbb{R}^n_+ & & s \in \mathbb{R}^n_+
\end{array}
\tag{LP}
$$

Where $x \in R^n$, $y \in \mathbb{R}^m$, $s \in \mathbb{R}^n$ are 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. Problems of linear programming can be introduced in various forms (including $\geq, \leq$ inequalities, free variables, possibly some linear fractions in objective) but they all can be transformed to 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

SOCP is another convex optimization problem which can be solved with great efficiency by interior point methods. Let us first define second order cone.

Definition 1.3 (Second order cone)**.** We say Q_n is second order cone of dimension n if

$$Q_n = \{x = (x_0, \bar{x}) \in \mathbb{R}^n |\ ||\bar{x}|| \leq x_0\},$$

where $||.||$ is standard euclidean norm and $x_0 \in \mathbb{R}$, $\bar{x} \in \mathbb{R}^{n-1}$.

One can define SOCP problem as LP with additional second order cone constraints.

Definition 1.4 (SOCP)**.** We say minimization problem is Second Order Cone Program (or shortly SOCP) if it has form of

$$
\begin{array}{ll|ll}
\multicolumn{2}{c|}{\textit{Primal}} & \multicolumn{2}{c}{\textit{Dual}} \\
\text{minimize} & c^T x & \text{maximize} & b^T y \\
\text{subject to} & Ax = b & \text{subject to} & A^T y + s = c \\
& x \in Q_n & & s \in Q_n
\end{array}
\tag{SOCP}
$$

Where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ and $s \in \mathbb{R}^n$ are 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_n are given problem data.

Relation to previous classes: Second Order Cone Programming includes convex LP as special case (in case of LP, Q_n is nonegative orthant). SOCP also includes convex QP. 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, $k = 0, \dots, m$ are positive semidefinite.

$$\begin{array}{ll} \text{minimize} & x^T P_0 x + q_0^T x + r_0 \\ \text{subject to} & x^T P_k x + q_k^T x + r_k \leq 0,\ (k = 1, \dots, m) \end{array} \tag{1.1}$$

First of all, we can get rid of quadratic objective function by introducing new variable $t \in \mathbb{R}$ and constraint $x^T P_0 x + q_0^T x + r_0 \leq t$ and then rewrite the whole program in variable $(t, x) \in \mathbb{R}^{n+1}$, with objective minimize t. To avoid tedious notation WLOG 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.

We will omit indices for simplicity and transform each convex quadratic constraint

$$x^T P x + q^T x + r \leq 0 \tag{1.2}$$

into second order cone constraint. Suppose that rank $P = h \geq 1$. 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 (1.2) as

$$(L^T x)^T (L^T x) \leq -q^T x - r. \tag{1.3}$$

It is known and also easily verified that $w \in \mathbb{R}^t$, $\xi \in \mathbb{R}$ and $\eta \in \mathbb{R}$ satisfy

$$w^T w \leq \xi\eta, \quad \xi \geq 0 \quad \text{and} \quad \eta \geq 0$$

if and only if they satisfy

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

If we take $w = L^T x,\ \xi = 1$ and $\eta = -q^T x - r$ we can convert inequality (1.3) into

$$\begin{pmatrix} v_0 \\ v \end{pmatrix} = \begin{pmatrix} 1 - q^T x - r \\ 1 + q^T x + r \\ 2L^T x \end{pmatrix} \in \mathbb{R}^{h+2} \text{ and } ||v|| \leq v_0. \tag{1.4}$$

Which is a second order cone constraint. Now the intersection of all such constraints is again second order cone, thus we have obtained problem of SOCP.

1.1.3 Semidefinite programming

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

Definition 1.5. 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 1.6 (SDP)**.** We say minimization problem is Semidefinite program (or shortly SDP) if it has form

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

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 matrices $A_0, A_1, \ldots, A_m \in \mathbb{S}^n$ and scalars $b_1, \ldots, b_m \in \mathbb{R}$ are given.

Relation to previous classes: The SDP primal-dual pair looks suspiciously similar to both pairs 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 further generalizes variable space from real vectors to symmetric matrices X, therefore we also need to use different notation in the linear objective and linear constraints. The cone constraint is provided by 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

$$x \in Q_n \;\Leftrightarrow\; ||\bar{x}|| \leq x_1 \;\Leftrightarrow \left\{ \begin{array}{c} \bar{x}^T \bar{x} \leq x_1^2 \\ 0 \leq x_1 \end{array} \right\} \tag{1.5}$$

$$\Leftrightarrow \left\{ \begin{array}{c} \frac{\bar{x}^T \bar{x}}{x_1} \leq x_1 \\ 0 \leq x_1 \end{array} \right\} \Leftrightarrow \begin{pmatrix} x_1 & \bar{x}^T \\ \bar{x} & x_1 I_{n-1} \end{pmatrix} \succeq 0. \tag{1.6}$$

Where last equivalence is provided by Schur complement lemma 2.1. It can be rewritten as linear matrix inequality (LMI)

$$x_1 I_n + \sum_{k=2}^{n} x_k E_{1k} \succeq 0 \tag{1.7}$$

The linear constraint $Ax = b$ can be easily cast to LMI form with

$$Ax = b \;\Leftrightarrow\; diag(Ax - b, b - Ax) \succeq 0, \tag{1.8}$$

where $diag(v)$ is the diagonal matrix with entries of v on the main diagonal. The resulting 2 LMIs can be simply combined into single one as a block diagonal matrix. The resulting SDP is then

$$\begin{aligned}
&\text{maximize} \;\; c^T x \\
&\text{s.t.} \;\; diag\begin{pmatrix} 0 \\ Ax - b \\ b - Ax \end{pmatrix} + x_1 diag\begin{pmatrix} I_n \\ 0 \\ 0 \end{pmatrix} + \textstyle\sum_{k=2}^{n} x_k diag\begin{pmatrix} E_{1k} \\ 0 \\ 0 \end{pmatrix} \succeq 0.
\end{aligned} \tag{1.9}$$

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.

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.

With this in mind we can generalize LP, SOCP, SDP together by single class.

Definition 1.7 (Conic Programming)**.** We say minimization problem is general Conic Program if it has form of

$$\begin{array}{ll|ll}
\multicolumn{2}{c|}{\textit{Primal}} & \multicolumn{2}{c}{\textit{Dual}} \\
\text{minimize} & c^T x & \text{maximize} & b^T y \\
\text{subject to} & Ax = b & \text{subject to} & A^T y + s = c \\
 & x \in \mathcal{K} & & s \in \mathcal{K}^*
\end{array} \tag{Conic Program}$$

Where $x \in \mathbb{R}^N$, $y \in \mathbb{R}^M$ and $s \in \mathbb{R}^N$ are 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 proper cone $\mathcal{K}$ are given problem data.

Conic programming contains any problems combined from LP, SOCP and SDP programs. If for the variable space we take a cartesian product of variables in these various programs and we take $\mathcal{K}$ as a cartesian product of the cones in their conic constraints.

QUESTION: is there any conic program which cannot be written as SDP?

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

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \in \mathcal{K} \end{array} \tag{1.10}$$

The Lagrangian of the problem is given by

$$\mathcal{L}(x, y, s) = c^T x + y^T (b - Ax) - s^T x, \quad \text{where } s \in \mathcal{K}^* \tag{1.11}$$

The last term is added to take account of the conic constraint $x \in \mathcal{K}$. From the very definition of dual cone:

$$\max_{s \in \mathcal{K}^*} -s^T x = \begin{cases} 0 & \text{if } x \in \mathcal{K}, \\ +\infty & \text{otherwise.} \end{cases} \tag{1.12}$$

The Lagrange dual function is

$$\begin{aligned} g(y, s) &= \min_x \ \mathcal{L}(x, y, s) && (1.13) \\ &= \min_x \ y^T b + (c + A^T y - s)^T x && (1.14) \\ &= \begin{cases} b^T y & \text{if } c - A^T y - s = 0, \\ -\infty & \text{otherwise.} \end{cases} && (1.15) \end{aligned}$$

Hence, dual for the problem is

$$\begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to} & A^T y + s = c \\ & s \in \mathcal{K}^* \end{array} \tag{1.16}$$

Substition of appropriate cone (nonegative orthant, second order cone or semidefinite cone) as $\mathcal{K}$ yelds earlier mentioned dual forms of the LP, SOCP and SDP.

1.2.2 Duality in QCQP

We will derive dual form of standard QCQP

$$\begin{array}{ll} \text{minimize} & x^T P_0 x + q_0^T x + r_0 \\ \text{subject to} & x^T P_k x + q_k^T x + r_k \le 0, \ (k = 1, \ldots, m) \end{array} \tag{1.17}$$

The Lagrangian of the problem is given by

$$\begin{aligned} \mathcal{L}(x,y) &= x^T P_0 x + q_0^T x + r_0 + \sum_{k=1}^{m} y_k (x^T P_k x + q_k^T x + r_k) && (1.18)\\ &= x^T P(y) x + q(y)^T x + r(y), && (1.19) \end{aligned}$$

where $y \geq 0$ and

$$P(y) = P_0 + \sum_{k=1}^{m} y_k P_k, \quad q(y) = q_0 + \sum_{k=1}^{m} y_k q_k, \quad r(y) = r_0 + \sum_{k=1}^{m} y_k r_k. \tag{1.20}$$

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

$$\begin{aligned} g(y) &= \min_x \ \mathcal{L}(x,y) && (1.21)\\ &= \begin{cases} -\frac{1}{4} q(y)^T P(y)^{\dagger} q(y) + r(y) & \text{if } P(y) \succeq 0, \ \ q(y) \in \mathcal{R}(P(y)) \\ -\infty & \text{otherwise.} \end{cases} && (1.22) \end{aligned}$$

Where $P^{\dagger}$ denotes pseudoinverse of P. 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 $P_0, P_1 \dots, P_m$, $q_0, q_1 \dots, q_m$, $r_0, r_1 \dots, r_m$ 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 and additional constraint.

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & t \leq -\frac{1}{4} q(y)^T P(y)^{\dagger} q(y) + r(y) \\ & y \geq 0 \\ & P(y) \succeq 0 \\ & \mathcal{R}(q(y)) \subseteq \mathcal{R}(P(y)). \end{array} \tag{1.23}$$

Thanks to Schur complement lemma 2.1 the above is equivalent to

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & M := \begin{pmatrix} r(y) - t & \frac{1}{2} q(y)^T \\ \frac{1}{2} q(y) & P(y) \end{pmatrix} \succeq 0, \\ & y \geq 0 \end{array} \tag{1.24}$$

Where the matrix M is easily expanded as

$$M = M(t) + \sum_{k=1}^{n} M(y_k); \tag{1.25}$$

Where

$$M(t) = \begin{pmatrix} -t & 0 \\ 0 & 0 \end{pmatrix}; \quad M(y_k) = \begin{pmatrix} r(y_k) & \frac{1}{2}q(y_k)^T \\ \frac{1}{2}q(y_k) & P(y_k) \end{pmatrix}. \tag{1.26}$$

1.3 Relaxations

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

Definition 1.8 (Relaxation)**.** Relaxation of minimization problem $z = \min\{c(x)|x \in X \subseteq \mathbb{R}^n\}$ is another minimization problem $z_R = \min\{c_R(x)|x \in X_R \subseteq \mathbb{R}^n\}$, with properties: $X \subseteq X_R$ and $c_R(x) \leq c(x) \forall x \in X$.

It easily follows, that solving relaxed problem will provide a lower bound on the optimal value of original problem. Sometimes we can from relaxation extract also some feasible solution of original problem and obtain upper bound for optimal value. Moreover, these bounds may not only give us an idea about the optimal value, but can provide a means to find optimal solution of original problem.

1.4 SDP relaxation

Consider QCQP (QCQP). Using identity $x^T A_k x = A_k \bullet xx^T$ it can be rewritten as follows

$$\begin{array}{ll} \text{minimize} & A_0 \bullet X + a_0^T x \\ \text{subject to} & A_k \bullet X + a_k^T x + \alpha_k \leq 0, \ (k = 1, \dots, m) \\ & X = xx^T \end{array} \tag{1.27}$$

Since matrices A_k $k = 0, \dots, m$ are symmetric, we can WLOG suppose that X is also symmetric. In order to get form similar to SDP (SDP), we need to hide linear terms $a_k^T x$. This can be done by lifting problem into higher dimension. Introducing $n+1 \times n+1$ matrices

$$M_k = \begin{pmatrix} \alpha_k & \frac{1}{2}a_k^T \\ \frac{1}{2}a_k & A_k \end{pmatrix}, \qquad Y = \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix},$$

where $k = 0, \ldots, m$, $\alpha_0 = 0$ and $X = xx^T$. We also suppose that $Y \in S^{n+1}$ (since we suppose $X \in S^n$). Using this notation (1.27) can be written as

$$\begin{array}{ll} \text{minimize} & M_0 \bullet Y \\ \text{subject to} & M_k \bullet Y \leq 0, \ (k = 1, \ldots, m) \\ & X = xx^T \end{array} \tag{1.28}$$

This problem looks almost like SDP but the last constraint is not quite what we need. Following lemma shows how to write it in more suitable form.

Lemma 1.1. *Let us have a vector $x \in \mathbb{R}^n$, $n \times n$ 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. Part (i) follows from Schur complement lemma for PSD (see Theorem 2.1 in appendices) To apply the this theorem we only need symmetry of Y and $1 \succ 0$ which holds.

Part (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 first (nonzero) row $(1, \ x^T)$. To match the first column it must be that $(i+1)$-th row of Y is $x_i(1, \ x^T)$, for $i = 1, \ldots, n$. Therefore, $X = xx^T$. $\square$

Remark. Notice that we have proven last implication of (ii) without using $Y \succeq 0$. In fact it is redundant. It also holds that $X = xx^T \Leftrightarrow \operatorname{rank} Y = 1$. 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.

Using the lemma and relaxing the nonconvex rank 1 constrain we obtain SDP relaxation of (QCQP)

$$\begin{array}{ll} \text{minimize} & M_0 \bullet Y \\ \text{subject to} & M_k \bullet Y \leq 0, \ (k = 1, \ldots, m) \\ & Y \succeq 0. \end{array} \tag{1.29}$$

Or expanding terms $M_k \bullet Y$ to see the original problem variable and data we have

$$\begin{array}{ll} \text{minimize} & A_0 \bullet X + a_0^T x \\ \text{subject to} & A_k \bullet X + a_k^T x + \alpha_k \le 0, \ (k = 1, \ldots, m) \\ & X \succeq xx^T. \end{array} \tag{1.30}$$

Looking back to definition of relaxation we see, that above problem is not a relaxation of QCQP, because it has different variable space $(X, x) \in \mathbb{R}^{n \times n} \times \mathbb{R}^n$ instead of $x \in \mathbb{R}^n$. However, it can be resolved by constructing equivalent problem to (1.27) by adding new redundant variable and constraint.

Proposition 1. *Problem (1.30) is relaxation of QCQP (QCQP) with additional varibale X and additional constraint $X = xx^T$ (wich is equivalent to original QCQP).*

Proof. Described QCQP with additional variable and constraint can be reformulated as (1.27). Comparing with SDP relaxation (1.30) these problems are equal up to the last constraint where $X = xx^T$ is replaced by $X \succeq xx^T$. The first one implies the second, because

$$X = xx^T \ \Rightarrow \ X - xx^T = 0 \succeq 0 \ \Rightarrow X \succeq xx^T.$$

Hence feasible set of (1.27) is subset of feasible set of (1.30). Same objective function then implies the second property of relaxation. □

Semidefinite program relaxations can provide tight bounds, but they can also be expensive to solve by classical interior point methods. 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

$$\begin{array}{ll} \text{minimize} & x^T B_0 x + b_0^T x \\ \text{subject to} & x^T B_k x + b_k^T x + \beta_i \le 0, \ (k = 1, \ldots, l) \end{array} \tag{1.31}$$

where all $B_k \succeq 0$ for $k = 0, \ldots, l$. We say that (1.31) is SOCP relaxation of (QCQP) if that x is feasible for (QCQP) implies that x is feasible for (1.31) 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 (1.31) is equivalent to the SDP

$$\begin{array}{ll} \text{minimize} & B_0 \bullet X + b_0^T x \\ \text{subject to} & B_k \bullet X + b_k^T x + \beta_i \leq 0, \ (k = 1, \ldots, l) \\ & X \succeq xx^T. \end{array} \tag{1.32}$$

In [4] authors provided SOCP relaxation of QCQP (QCQP) 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

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

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

$$\begin{array}{rl} x^T A_k^+ x + a_k^T x + \alpha_k & \leq z_k \\ x^T A_k^- x & \leq z_k. \end{array} \tag{1.33}$$

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.

1.5.1 SOCP relaxation of semidefinite constraint

Semidefinite constraint $X - xx^T \succeq 0$ in (1.30) is equivalent to $C \bullet (X - xx^T) \geq 0$ for all $C \in S_+^n$. Using this, in [4] authors propose SOCP relaxation of the semidefinite constraint $X - xx^T \succeq 0$ by replacing it with multiple constraints of the form

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

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

$$\begin{pmatrix} v_0^i \\ v^i \end{pmatrix} = \begin{pmatrix} 1 + C_i \bullet X \\ 1 - C_i \bullet X \\ 2L_i^T x \end{pmatrix}, \quad ||v^i|| \leq v_0^i, \quad (i = 1, \ldots, l) \tag{1.34}$$

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 (QCQP) somewhere between SDP and SOCP. We will describe their approach on specific case they provide as gentle introduction. We are dealing with (QCQP)

$$\begin{array}{ll} \text{minimize} & x^T A_0 x + a_0^T x \\ \text{subject to} & x^T A_k x + a_k^T x + \alpha_k \leq 0, \ (k = 1, \ldots, m) \end{array} \tag{1.35}$$

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

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

which has following SOCP-SDP relaxation

$$\begin{array}{ll} \text{minimize} & -y_0 Tr(X) + x^T(A_0 + y_0 I)x + a_0^T x \\ \text{subject to} & -y_k Tr(X) + x^T(A_k + y_k I)x + a_k^T x + \alpha_k \leq 0, \\ & (k = 1, \ldots, m) \\ & X \succeq xx^T \end{array} \tag{1.36}$$

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

Proposition 2. *Given a vector x and scalars $X_{11}, \ldots, X_{nn}$, there exists a symmetric-matrix completion $X \in S^n$ of $X_{11}, \ldots, X_{nn}$ satisfying $X \succeq xx^T$ if and only if $X_{jj} \geq x_j^2$ for all $j = 1, \ldots, n$.* [10]

Thus, in light of this proposition, the problem with additional bounding constraints $X_{jj} \leq x_j$, the problem (1.36) is equivalent to

$$\begin{array}{ll} \text{minimize} & -y_0 Tr(X) + x^T(A_0 + y_0 I)x + a_0^T x \\ \text{subject to} & -y_k Tr(X) + x^T(A_k + y_k I)x + a_k^T x + \alpha_k \leq 0, \\ & (k = 1, \ldots, m) \\ & x_j^2 \leq X_{jj} \leq x_j \quad (j = 1, \ldots, n) \end{array} \tag{1.37}$$

Compared to SDP relaxation (1.30), which has $O(n^2)$ variables, problem (1.37) 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 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.

Chapter 2

Appendices

Theorem 2.1 (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. [9, 5]

Proof. (will be added later)

2.1 Cones

Definition 2.1 (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 2.2 (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 2.3 (Dual cone)**.** For any cone $\mathcal{K}$ we define its dual cone $\mathcal{K}^*$ as a set

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

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

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

[8] 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`

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

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