Introduction to Semidefinite Programming (SDP)

Robert M. Freund

March, 2004

1 Introduction

Semidefinite programming (SDP) is the most exciting development in mathematical programming in the 1990's. SDP has applications in such diverse fields as traditional convex constrained optimization, control theory, and combinatorial optimization. Because SDP is solvable via interior point methods, most of these applications can usually be solved very efficiently in practice as well as in theory.

2 Review of Linear Programming

Consider the linear programming problem in standard form:

$$\begin{array}{lll} LP: & \text{minimize} & c \cdot x \\ & \text{s.t.} & a_i \cdot x = b_i, \quad i = 1, \ldots, m \\ & & x \in \Re^n_+. \end{array}$$

Here x is a vector of n variables, and we write "$c \cdot x$" for the inner-product "$\sum_{j=1}^n c_j x_j$", etc.

Also, $\Re^n_+ := \{x \in \Re^n \mid x \geq 0\}$, and we call $\Re^n_+$ the nonnegative orthant. In fact, $\Re^n_+$ is a *closed convex cone,* where K is called a closed a convex cone if K satisfies the following two conditions:

- If $x, w \in K$, then $\alpha x + \beta w \in K$ for all nonnegative scalars α and β.
- K is a closed set.

In words, LP is the following problem:

"Minimize the linear function $c \cdot x$, subject to the condition that x must solve m given equations $a_i \cdot x = b_i, i = 1, \ldots, m$, and that x must lie in the closed convex cone $K = \Re^n_+$."

We will write the standard linear programming dual problem as:

$$
\begin{array}{lll}
LD: & \text{maximize} & \sum_{i=1}^{m} y_i b_i \\
 & \text{s.t.} & \sum_{i=1}^{m} y_i a_i + s = c \\
 & & s \in \Re^n_+ .
\end{array}
$$

Given a feasible solution x of LP and a feasible solution (y, s) of LD, the duality gap is simply $c \cdot x - \sum_{i=1}^{m} y_i b_i = (c - \sum_{i=1}^{m} y_i a_i) \cdot x = s \cdot x \geq 0$, because $x \geq 0$ and $s \geq 0$. We know from LP duality theory that so long as the primal problem LP is feasible and has bounded optimal objective value, then the primal and the dual both attain their optima with no duality gap. That is, there exists x^* and (y^*, s^*) feasible for the primal and dual, respectively, such that $c \cdot x^* - \sum_{i=1}^{m} y_i^* b_i = s^* \cdot x^* = 0$.

3 Facts about Matrices and the Semidefinite Cone

If X is an $n \times n$ matrix, then X is a positive semidefinite (psd) matrix if

$$v^T X v \geq 0 \quad \text{for any} \quad v \in \Re^n .$$

If X is an $n \times n$ matrix, then X is a positive definite (pd) matrix if

$$v^T X v > 0 \text{ for any } v \in \Re^n, v \neq 0.$$

Let S^n denote the set of symmetric $n \times n$ matrices, and let S^n_+ denote the set of positive semidefinite (*psd*) $n \times n$ symmetric matrices. Similarly let S^n_{++} denote the set of positive definite (*pd*) $n \times n$ symmetric matrices.

Let X and Y be any symmetric matrices. We write "$X \succeq 0$" to denote that X is symmetric and positive semidefinite, and we write "$X \succeq Y$" to denote that $X - Y \succeq 0$. We write "$X \succ 0$" to denote that X is symmetric and positive definite, etc.

Remark 1 *$S^n_+ = \{X \in S^n \mid X \succeq 0\}$ is a closed convex cone in $\Re^{n^2}$ of dimension $n \times (n+1)/2$.*

To see why this remark is true, suppose that $X, W \in S^n_+$. Pick any scalars $\alpha, \beta \geq 0$. For any $v \in \Re^n$, we have:

$$v^T(\alpha X + \beta W)v = \alpha v^T X v + \beta v^T W v \geq 0,$$

whereby $\alpha X + \beta W \in S^n_+$. This shows that S^n_+ is a cone. It is also straightforward to show that S^n_+ is a closed set.

Recall the following properties of symmetric matrices:

- If $X \in S^n$, then $X = QDQ^T$ for some orthonormal matrix Q and some diagonal matrix D. (Recall that Q is orthonormal means that

$Q^{-1} = Q^T$, and that D is diagonal means that the off-diagonal entries of D are all zeros.)

- If $X = QDQ^T$ as above, then the columns of Q form a set of n orthogonal eigenvectors of X, whose eigenvalues are the corresponding diagonal entries of D.

- $X \succeq 0$ if and only if $X = QDQ^T$ where the eigenvalues (i.e., the diagonal entries of D) are all nonnegative.

- $X \succ 0$ if and only if $X = QDQ^T$ where the eigenvalues (i.e., the diagonal entries of D) are all positive.

- If $X \succeq 0$ and if $X_{ii} = 0$, then $X_{ij} = X_{ji} = 0$ for all $j = 1, \ldots, n$.

- Consider the matrix M defined as follows:

$$M = \begin{pmatrix} P & v \\ v^T & d \end{pmatrix},$$

  where $P \succ 0$, v is a vector, and d is a scalar. Then $M \succ 0$ if and only if $d - v^T P^{-1} v > 0$.

4 Semidefinite Programming

Let $X \in S^n$. We can think of X as a matrix, or equivalently, as an array of n^2 components of the form $(x_{11}, \ldots, x_{nn})$. We can also just think of X as an object (a vector) in the space S^n. All three different equivalent ways of looking at X will be useful.

What will a linear function of X look like? If $C(X)$ is a linear function of X, then $C(X)$ can be written as $C \bullet X$, where

$$C \bullet X := \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij}.$$

If X is a symmetric matrix, there is no loss of generality in assuming that the matrix C is also symmetric. With this notation, we are now ready to define a semidefinite program. A semidefinite program (SDP) is an optimization problem of the form:

$$\begin{array}{llll} SDP: & \text{minimize} & C \bullet X & \\ & \text{s.t.} & A_i \bullet X = b_i & , i = 1, \ldots, m, \\ & & X \succeq 0, & \end{array}$$

Notice that in an SDP that the variable is the matrix X, but it might be helpful to think of X as an array of n^2 numbers or simply as a vector in S^n. The objective function is the linear function $C \bullet X$ and there are m linear equations that X must satisfy, namely $A_i \bullet X = b_i \quad , i = 1, \ldots, m$. The variable X also must lie in the (closed convex) cone of positive semidefinite symmetric matrices S^n_+. Note that the data for SDP consists of the symmetric matrix C (which is the data for the objective function) and the m symmetric matrices $A_1, \ldots, A_m$, and the $m-$vector b, which form the m linear equations.

Let us see an example of an SDP for $n = 3$ and $m = 2$. Define the following matrices:

$$A_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 3 & 7 \\ 1 & 7 & 5 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 2 & 8 \\ 2 & 6 & 0 \\ 8 & 0 & 4 \end{pmatrix}, \quad \text{and} \quad C = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 9 & 0 \\ 3 & 0 & 7 \end{pmatrix},$$

and $b_1 = 11$ and $b_2 = 19$. Then the variable X will be the 3×3 symmetric matrix:

$$X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix},$$

and so, for example,

$$\begin{aligned} C \bullet X &= x_{11} + 2x_{12} + 3x_{13} + 2x_{21} + 9x_{22} + 0x_{23} + 3x_{31} + 0x_{32} + 7x_{33} \\ &= x_{11} + 4x_{12} + 6x_{13} + 9x_{22} + 0x_{23} + 7x_{33}. \end{aligned}$$

since, in particular, X is symmetric. Therefore the SDP can be written as:

$$\begin{array}{lll} SDP: & \text{minimize} & x_{11} + 4x_{12} + 6x_{13} + 9x_{22} + 0x_{23} + 7x_{33} \\ & \text{s.t.} & \\ & & x_{11} + 0x_{12} + 2x_{13} + 3x_{22} + 14x_{23} + 5x_{33} \;=\; 11 \\ & & 0x_{11} + 4x_{12} + 16x_{13} + 6x_{22} + 0x_{23} + 4x_{33} \;=\; 19 \\ & & X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} \succeq 0. \end{array}$$

Notice that SDP looks remarkably similar to a linear program. However, the standard LP constraint that x must lie in the nonnegative orthant is replaced by the constraint that the variable X must lie in the cone of positive semidefinite matrices. Just as "$x \geq 0$" states that each of the n components of x must be nonnegative, it may be helpful to think of "$X \succeq 0$" as stating that each of the n *eigenvalues* of X must be nonnegative.

It is easy to see that a linear program LP is a special instance of an SDP. To see one way of doing this, suppose that $(c, a_1, \ldots, a_m, b_1, \ldots, b_m)$ comprise the data for LP. Then define:

$$A_i = \begin{pmatrix} a_{i1} & 0 & \ldots & 0 \\ 0 & a_{i2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & a_{in} \end{pmatrix}, \quad i = 1, \ldots, m, \quad \text{and} \quad C = \begin{pmatrix} c_1 & 0 & \ldots & 0 \\ 0 & c_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & c_n \end{pmatrix}.$$

Then LP can be written as:

$$\begin{array}{lll} SDP: & \text{minimize} & C \bullet X \\ & & \\ & \text{s.t.} & A_i \bullet X = b_i \ \ , i = 1, \ldots, m, \\ & & X_{ij} = 0, \quad i = 1, \ldots, n, \quad j = i+1, \ldots, n, \\ & & X \succeq 0, \end{array}$$

with the association that

$$X = \begin{pmatrix} x_1 & 0 & \ldots & 0 \\ 0 & x_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & x_n \end{pmatrix}.$$

Of course, in practice one would never *want* to convert an instance of LP into an instance of SDP. The above construction merely shows that SDP

includes linear programming as a special case.

5 Semidefinite Programming Duality

The dual problem of SDP is defined (or derived from first principles) to be:

$$\begin{array}{lll} SDD: & \text{maximize} & \sum\limits_{i=1}^{m} y_i b_i \\ & \text{s.t.} & \sum\limits_{i=1}^{m} y_i A_i + S = C \\ & & S \succeq 0. \end{array}$$

One convenient way of thinking about this problem is as follows. Given multipliers $y_1, \ldots, y_m$, the objective is to maximize the linear function $\sum_{i=1}^{m} y_i b_i$. The constraints of SDD state that the matrix S defined as

$$S = C - \sum_{i=1}^{m} y_i A_i$$

must be positive semidefinite. That is,

$$C - \sum_{i=1}^{m} y_i A_i \succeq 0.$$

We illustrate this construction with the example presented earlier. The dual problem is:

$$
\begin{array}{lll}
SDD: & \text{maximize} & 11y_1 + 19y_2 \\[2ex]
 & \text{s.t.} & y_1 \begin{pmatrix} 1 & 0 & 1 \\ 0 & 3 & 7 \\ 1 & 7 & 5 \end{pmatrix} + y_2 \begin{pmatrix} 0 & 2 & 8 \\ 2 & 6 & 0 \\ 8 & 0 & 4 \end{pmatrix} + S = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 9 & 0 \\ 3 & 0 & 7 \end{pmatrix} \\[2ex]
 & & S \succeq 0,
\end{array}
$$

which we can rewrite in the following form:

$$
\begin{array}{ll}
SDD: \quad \text{maximize} & 11y_1 + 19y_2 \\[2ex]
\qquad \text{s.t.} & \\
 & \begin{pmatrix} 1 - 1y_1 - 0y_2 & 2 - 0y_1 - 2y_2 & 3 - 1y_1 - 8y_2 \\ 2 - 0y_1 - 2y_2 & 9 - 3y_1 - 6y_2 & 0 - 7y_1 - 0y_2 \\ 3 - 1y_1 - 8y_2 & 0 - 7y_1 - 0y_2 & 7 - 5y_1 - 4y_2 \end{pmatrix} \succeq 0.
\end{array}
$$

It is often easier to "see" and to work with a semidefinite program when it is presented in the format of the dual SDD, since the variables are the m multipliers $y_1, \ldots, y_m$.

As in linear programming, we can switch from one format of SDP (primal or dual) to any other format with great ease, and there is no loss of generality in assuming a particular specific format for the primal or the dual.

The following proposition states that weak duality must hold for the primal and dual of SDP:

Proposition 5.1 *Given a feasible solution X of SDP and a feasible solution (y, S) of SDD, the duality gap is $C \bullet X - \sum_{i=1}^{m} y_i b_i = S \bullet X \geq 0$. If $C \bullet X - \sum_{i=1}^{m} y_i b_i = 0$, then X and (y, S) are each optimal solutions to SDP and SDD, respectively, and furthermore, $SX = 0$.*

In order to prove Proposition 5.1, it will be convenient to work with the *trace* of a matrix, defined below:

$$\operatorname{trace}(M) = \sum_{j=1}^{n} M_{jj}.$$

Simple arithmetic can be used to establish the following two elementary identities:

- $\operatorname{trace}(MN) = \operatorname{trace}(NM)$
- $A \bullet B = \operatorname{trace}(A^T B)$

Proof of Proposition 5.1. For the first part of the proposition, we must show that if $S \succeq 0$ and $X \succeq 0$, then $S \bullet X \geq 0$. Let $S = PDP^T$ and $X = QEQ^T$ where P, Q are orthonormal matrices and D, E are nonnegative diagonal matrices. We have:

$$S \bullet X = \operatorname{trace}(S^T X) = \operatorname{trace}(SX) = \operatorname{trace}(PDP^T QEQ^T)$$

$$= \operatorname{trace}(DP^T QEQ^T P) = \sum_{j=1}^{n} D_{jj}(P^T QEQ^T P)_{jj} \geq 0,$$

where the last inequality follows from the fact that all $D_{jj} \geq 0$ and the fact that the diagonal of the symmetric positive semidefinite matrix $P^T QEQ^T P$ must be nonnegative.

To prove the second part of the proposition, suppose that $\text{trace}(SX) = 0$. Then from the above equalities, we have

$$\sum_{j=1}^{n} D_{jj}(P^T QEQ^T P)_{jj} = 0.$$

However, this implies that for each $j = 1, \ldots, n$, either $D_{jj} = 0$ or the $(P^T QEQ^T P)_{jj} = 0$. Furthermore, the latter case implies that the j^{th} row of $P^T QEQ^T P$ is all zeros. Therefore $DP^T QEQ^T P = 0$, and so $SX = PDP^T QEQ^T = 0$.
q.e.d.

Unlike the case of linear programming, we cannot assert that either SDP or SDD will attain their respective optima, and/or that there will be no duality gap, unless certain regularity conditions hold. One such regularity condition which ensures that strong duality will prevail is a version of the "Slater condition", summarized in the following theorem which we will not prove:

Theorem 5.1 *Let z_P^* and z_D^* denote the optimal objective function values of SDP and SDD, respectively. Suppose that there exists a feasible solution $\hat{X}$ of SDP such that $\hat{X} \succ 0$, and that there exists a feasible solution $(\hat{y}, \hat{S})$ of SDD such that $\hat{S} \succ 0$. Then both SDP and SDD attain their optimal values, and $z_P^* = z_D^*$.*

6 Key Properties of Linear Programming that do not extend to SDP

The following summarizes some of the more important properties of linear programming that do not extend to SDP:

- There may be a finite or infinite duality gap. The primal and/or dual may or may not attain their optima. As noted above in Theorem 5.1, both programs will attain their common optimal value if both programs have feasible solutions in the interior of the semidefinite cone.
- There is no finite algorithm for solving SDP. There is a simplex algorithm, but it is not a finite algorithm. There is no direct analog of a "basic feasible solution" for SDP.
- Given rational data, the feasible region may have no rational solutions. The optimal solution may not have rational components or rational eigenvalues.
- Given rational data whose binary encoding is size L, the norms of any feasible and/or optimal solutions may exceed 2^{2^L} (or worse).
- Given rational data whose binary encoding is size L, the norms of any feasible and/or optimal solutions may be less than 2^{-2^L} (or worse).

7 SDP in Combinatorial Optimization

SDP has wide applicability in combinatorial optimization. A number of $NP-$hard combinatorial optimization problems have convex relaxations that are semidefinite programs. In many instances, the SDP relaxation is very tight in practice, and in certain instances in particular, the optimal solution to the SDP relaxation can be converted to a feasible solution for the original problem with provably good objective value. An example of the use of SDP in combinatorial optimization is given below.

7.1 An SDP Relaxation of the MAX CUT Problem

Let G be an undirected graph with nodes $N = \{1, \ldots, n\}$, and edge set E. Let $w_{ij} = w_{ji}$ be the weight on edge (i, j), for $(i, j) \in E$. We assume that $w_{ij} \geq 0$ for all $(i, j) \in E$. The MAX CUT problem is to determine a subset S of the nodes N for which the sum of the weights of the edges that cross from S to its complement $\bar{S}$ is maximized (where $(\bar{S} := N \setminus S)$.

We can formulate MAX CUT as an integer program as follows. Let $x_j = 1$ for $j \in S$ and $x_j = -1$ for $j \in \bar{S}$. Then our formulation is:

$$
\begin{array}{lll}
MAXCUT: & \text{maximize}_x & \frac{1}{4} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}(1 - x_i x_j) \\
& & \\
& \text{s.t.} & x_j \in \{-1, 1\}, \quad j = 1, \ldots, n.
\end{array}
$$

Now let

$$
Y = xx^T,
$$

whereby

$$
Y_{ij} = x_i x_j \quad i = 1, \ldots, n, \quad j = 1, \ldots, n.
$$

Also let W be the matrix whose $(i, j)^{\text{th}}$ element is w_{ij} for $i = 1, \ldots, n$ and $j = 1, \ldots, n$. Then MAX CUT can be equivalently formulated as:

$$
\begin{aligned}
MAXCUT: \quad \text{maximize}_{Y,x} \quad & \tfrac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} - W \bullet Y \\
\text{s.t.} \quad & x_j \in \{-1, 1\}, \quad j = 1, \ldots, n \\
& Y = xx^T.
\end{aligned}
$$

Notice in this problem that the first set of constraints are equivalent to $Y_{jj} = 1, j = 1, \ldots, n$. We therefore obtain:

$$
\begin{aligned}
MAXCUT: \quad \text{maximize}_{Y,x} \quad & \tfrac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} - W \bullet Y \\
\text{s.t.} \quad & Y_{jj} = 1, \quad j = 1, \ldots, n \\
& Y = xx^T.
\end{aligned}
$$

Last of all, notice that the matrix $Y = xx^T$ is a symmetric rank-1 positive semidefinite matrix. If we *relax* this condition by removing the rank-1 restriction, we obtain the following relaxtion of MAX CUT, which is a semidefinite program:

$$
\begin{aligned}
RELAX: \quad \text{maximize}_{Y} \quad & \tfrac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} - W \bullet Y \\
\text{s.t.} \quad & Y_{jj} = 1, \quad j = 1, \ldots, n \\
& Y \succeq 0.
\end{aligned}
$$

It is therefore easy to see that RELAX provides an upper bound on MAX-CUT, i.e.,

$$MAXCUT \leq RELAX.$$

As it turns out, one can also prove without too much effort that:

$$0.87856\ RELAX \leq MAXCUT \leq RELAX.$$

This is an impressive result, in that it states that the value of the semidefinite relaxation is guaranteed to be no more than 12% higher than the value of NP-hard problem MAX CUT.

8 SDP in Convex Optimization

As stated above, SDP has very wide applications in convex optimization. The types of constraints that can be modeled in the SDP framework include: linear inequalities, convex quadratic inequalities, lower bounds on matrix norms, lower bounds on determinants of symmetric positive semidefinite matrices, lower bounds on the geometric mean of a nonnegative vector, plus many others. Using these and other constructions, the following problems (among many others) can be cast in the form of a semidefinite program: linear programming, optimizing a convex quadratic form subject to convex quadratic inequality constraints, minimizing the volume of an ellipsoid that covers a given set of points and ellipsoids, maximizing the volume of an ellipsoid that is contained in a given polytope, plus a variety of maximum eigenvalue and minimum eigenvalue problems. In the subsections below we demonstrate how some important problems in convex optimization can be re-formulated as instances of SDP.

8.1 SDP for Convex Quadratically Constrained Quadratic Programming, Part I

A convex quadratically constrained quadratic program is a problem of the form:

$$
\begin{array}{lll}
QCQP: & \underset{x}{\text{minimize}} & x^T Q_0 x + q_0^T x + c_0 \\
 & \text{s.t.} & x^T Q_i x + q_i^T x + c_i \le 0 \quad , i = 1, \ldots, m,
\end{array}
$$

where the $Q_0 \succeq 0$ and $Q_i \succeq 0, \quad i = 1, \ldots, m$. This problem is the same as:

$$
\begin{array}{lll}
QCQP: & \underset{x,\theta}{\text{minimize}} & \theta \\
 & \text{s.t.} & x^T Q_0 x + q_0^T x + c_0 - \theta \le 0 \\
 & & x^T Q_i x + q_i^T x + c_i \le 0 \quad , i = 1, \ldots, m.
\end{array}
$$

We can factor each Q_i into

$$Q_i = M_i^T M_i$$

for some matrix M_i. Then note the equivalence:

$$
\left(\begin{array}{cc} I & M_i x \\ x^T M_i^T & -c_i - q_i^T x \end{array} \right) \succeq 0 \quad \Leftrightarrow \quad x^T Q_i x + q_i^T x + c_i \le 0.
$$

In this way we can write $QCQP$ as:

$$
\begin{array}{lll}
QCQP: & \underset{x,\theta}{\text{minimize}} & \theta \\
& \text{s.t.} &
\end{array}
$$

$$
\left(\begin{array}{cc} I & M_0 x \\ x^T M_0^T & -c_0 - q_0^T x + \theta \end{array} \right) \succeq 0
$$

$$
\left(\begin{array}{cc} I & M_i x \\ x^T M_i^T & -c_i - q_i^T x \end{array} \right) \succeq 0 \quad , i = 1, \ldots, m.
$$

Notice in the above formulation that the variables are θ and x and that all matrix coefficients are linear functions of θ and x.

8.2 SDP for Convex Quadratically Constrained Quadratic Programming, Part II

As it turns out, there is an alternative way to formulate $QCQP$ as a semi-definite program. We begin with the following elementary proposition.

Proposition 8.1 *Given a vector $x \in \Re^k$ and a matrix $W \in \Re^{k \times k}$, then*

$$
\left(\begin{array}{cc} 1 & x^T \\ x & W \end{array} \right) \succeq 0 \quad \text{if and only if} \quad W \succeq x x^T .
$$

Using this proposition, it is straightforward to show that $QCQP$ is equivalent to the following semi-definite program:

$$
\begin{array}{lll}
QCQP: & \underset{x,W,\theta}{\text{minimize}} & \theta \\
& \text{s.t.} &
\end{array}
$$

$$
\begin{pmatrix} c_0 - \theta & \frac{1}{2}q_0^T \\ \frac{1}{2}q_0 & Q_0 \end{pmatrix} \bullet \begin{pmatrix} 1 & x^T \\ x & W \end{pmatrix} \le 0
$$

$$
\begin{pmatrix} c_i & \frac{1}{2}q_i^T \\ \frac{1}{2}q_i & Q_i \end{pmatrix} \bullet \begin{pmatrix} 1 & x^T \\ x & W \end{pmatrix} \le 0 \quad , i = 1, \ldots, m
$$

$$
\begin{pmatrix} 1 & x^T \\ x & W \end{pmatrix} \succeq 0 \ .
$$

Notice in this formulation that there are now $\frac{(n+1)(n+2)}{2}$ variables, but that the constraints are all linear inequalities as opposed to semi-definite inequalities.

8.3 SDP for the Smallest Circumscribed Ellipsoid Problem

A given matrix $R \succ 0$ and a given point z can be used to define an ellipsoid in $\Re^n$:

$$
E_{R,z} := \{y \mid (y - z)^T R(y - z) \le 1\}.
$$

One can prove that the volume of $E_{R,z}$ is proportional to $\det(R^{-1})$.

Suppose we are given a convex set $X \in \Re^n$ described as the convex hull of k points $c_1, \ldots, c_k$. We would like to find an ellipsoid circumscribing these k points that has minimum volume. Our problem can be written in the following form:

$$\begin{array}{lll} MCP: & \underset{R,z}{\text{minimize}} & \text{vol}\,(E_{R,z}) \\ & \text{s.t.} & c_i \in E_{R,z}, \quad i = 1, \ldots, k, \end{array}$$

which is equivalent to:

$$\begin{array}{lll} MCP: & \underset{R,z}{\text{minimize}} & -\ln(\det(R)) \\ & \text{s.t.} & (c_i - z)^T R (c_i - z) \le 1, \quad i = 1, \ldots, k \\ & & \\ & & R \succ 0, \end{array}$$

Now factor $R = M^2$ where $M \succ 0$ (that is, M is a square root of R), and now MCP becomes:

$$\begin{array}{lll} MCP: & \underset{M,z}{\text{minimize}} & -\ln(\det(M^2)) \\ & \text{s.t.} & (c_i - z)^T M^T M (c_i - z) \le 1, \quad i = 1, \ldots, k, \\ & & \\ & & M \succ 0. \end{array}$$

Next notice the equivalence:

$$\begin{pmatrix} I & Mc_i - Mz \\ (Mc_i - Mz)^T & 1 \end{pmatrix} \succeq 0 \quad \Leftrightarrow \quad (c_i - z)^T M^T M (c_i - z) \le 1$$

In this way we can write MCP as:

$$
\begin{array}{lll}
MCP: & \underset{M,z}{\text{minimize}} & -2\ln(\det(M)) \\
& \text{s.t.} & \begin{pmatrix} I & Mc_i - Mz \\ (Mc_i - Mz)^T & 1 \end{pmatrix} \succeq 0, \quad i = 1, \ldots, k, \\
& & M \succ 0.
\end{array}
$$

Last of all, make the substitution $y = Mz$ to obtain:

$$
\begin{array}{lll}
MCP: & \underset{M,y}{\text{minimize}} & -2\ln(\det(M)) \\
& \text{s.t.} & \begin{pmatrix} I & Mc_i - y \\ (Mc_i - y)^T & 1 \end{pmatrix} \succeq 0, \quad i = 1, \ldots, k, \\
& & M \succ 0.
\end{array}
$$

Notice that this last program involves semidefinite constraints where all of the matrix coefficients are linear functions of the variables M and y. However, the objective function is not a linear function. It is possible to convert this problem further into a genuine instance of SDP, because there is a way to convert constraints of the form

$$-\ln(\det(X)) \le \theta$$

to a semidefinite system. Nevertheless, this is not necessary, either from a theoretical or a practical viewpoint, because it turns out that the function $f(X) = -\ln(\det(X))$ is extremely well-behaved and is very easy to optimize (both in theory and in practice).

Finally, note that after solving the formulation of MCP above, we can recover the matrix R and the center z of the optimal ellipsoid by computing

$$R = M^2 \quad \text{and} \quad z = M^{-1}y.$$

8.4 SDP for the Largest Inscribed Ellipsoid Problem

Recall that a given matrix $R \succ 0$ and a given point z can be used to define an ellipsoid in $\Re^n$:

$$E_{R,z} := \{x \mid (x - z)^T R(x - z) \leq 1\},$$

and that the volume of $E_{R,z}$ is proportional to $\det(R^{-1})$.

Suppose we are given a convex set $X \in \Re^n$ described as the intersection of k halfspaces $\{x \mid (a_i)^T x \leq b_i\}, i = 1, \ldots, k$, that is,

$$X = \{x \mid Ax \leq b\}$$

where the i^{th} row of the matrix A consists of the entries of the vector $a_i, i = 1, \ldots, k$. We would like to find an ellipsoid inscribed in X of maximum volume. Our problem can be written in the following form:

$$\begin{array}{lll} MIP: & \underset{R,z}{\text{maximize}} & \text{vol}\,(E_{R,z}) \\ & \text{s.t.} & E_{R,z} \subset X, \end{array}$$

which is equivalent to:

$$
\begin{array}{lll}
MIP: & \underset{R,z}{\text{maximize}} & \det(R^{-1}) \\
& \text{s.t.} & E_{R,z} \subset \{x \mid (a_i)^T x \le b_i\}, \quad i = 1, \ldots, k \\
& & \\
& & R \succ 0,
\end{array}
$$

which is equivalent to:

$$
\begin{array}{lll}
MIP: & \underset{R,z}{\text{maximize}} & \ln(\det(R^{-1})) \\
& \text{s.t.} & \max_x\{a_i^T x \mid (x-z)^T R(x-z) \le 1\} \le b_i, \quad i = 1, \ldots, k \\
& & \\
& & R \succ 0.
\end{array}
$$

For a given $i = 1, \ldots, k$, the solution to the optimization problem in the i^{th} constraint is

$$
x^* = z + \frac{R^{-1}a_i}{\sqrt{a_i^T R^{-1} a_i}}
$$

with optimal objective function value

$$
a_i^T z + \sqrt{a_i^T R^{-1} a_i} \quad ,
$$

and so MIP can be rewritten as:

$$
\begin{array}{llll}
MIP: & \underset{R,z}{\text{maximize}} & \ln(\det(R^{-1})) & \\
& \text{s.t.} & a_i^T z + \sqrt{a_i^T R^{-1} a_i} & \leq b_i, \quad i = 1, \ldots, k \\
& & R \succ 0. &
\end{array}
$$

Now factor $R^{-1} = M^2$ where $M \succ 0$ (that is, M is a square root of R^{-1}), and now MIP becomes:

$$
\begin{array}{llll}
MIP: & \underset{M,z}{\text{maximize}} & \ln(\det(M^2)) & \\
& \text{s.t.} & a_i^T z + \sqrt{a_i^T M^T M a_i} & \leq b_i, \quad i = 1, \ldots, k \\
& & M \succ 0, &
\end{array}
$$

which we can re-write as:

$$
\begin{array}{lll}
MIP: & \underset{M,z}{\text{maximize}} & 2\ln(\det(M)) \\
& \text{s.t.} & a_i^T M^T M a_i \leq (b_i - a_i^T z)^2, \quad i = 1, \ldots, k \\
& & b_i - a_i^T z \geq 0, \quad i = 1, \ldots, k \\
& & M \succ 0.
\end{array}
$$

Next notice the equivalence:

$$
\left(\begin{array}{cc} (b_i - a_i^T z) I & M a_i \\ (M a_i)^T & (b_i - a_i^T z) \end{array} \right) \succeq 0 \quad \Leftrightarrow \quad \left\{ \begin{array}{c} a_i^T M^T M a_i \leq (b_i - a_i^T z)^2 \\ \text{and} \\ b_i - a_i^T z \geq 0. \end{array} \right\}
$$

In this way we can write MIP as:

$$
\begin{array}{lll}
MIP: & \underset{M,z}{\text{minimize}} & -2\ln(\det(M)) \\
& \text{s.t.} & \begin{pmatrix} (b_i - a_i^T z)I & Ma_i \\ (Ma_i)^T & (b_i - a_i^T z) \end{pmatrix} \succeq 0, \quad i = 1, \ldots, k, \\
& & M \succ 0.
\end{array}
$$

Notice that this last program involves semidefinite constraints where all of the matrix coefficients are linear functions of the variables M and z. However, the objective function is not a linear function. It is possible, but not necessary in practice, to convert this problem further into a genuine instance of SDP, because there is a way to convert constraints of the form

$$
-\ln(\det(X)) \leq \theta
$$

to a semidefinite system. Such a conversion is not necessary, either from a theoretical or a practical viewpoint, because it turns out that the function $f(X) = -\ln(\det(X))$ is extremely well-behaved and is very easy to optimize (both in theory and in practice).

Finally, note that after solving the formulation of MIP above, we can recover the matrix R of the optimal ellipsoid by computing

$$
R = M^{-2}.
$$

8.5 SDP for Eigenvalue Optimization

There are many types of eigenvalue optimization problems that can be formualated as SDPs. A typical eigenvalue optimization problem is to create a matrix

$$S := B - \sum_{i=1}^{k} w_i A_i$$

given symmetric data matrices B and $A_i, i = 1, \ldots, k$, using weights $w_1, \ldots, w_k$, in such a way to minimize the difference between the largest and smallest eigenvalue of S. This problem can be written down as:

$$\begin{array}{rll} EOP: & \underset{w,S}{\text{minimize}} & \lambda_{\max}(S) - \lambda_{\min}(S) \\ & \text{s.t.} & S = B - \sum_{i=1}^{k} w_i A_i, \end{array}$$

where $\lambda_{\min}(S)$ and $\lambda_{\max}(S)$ denote the smallest and the largest eigenvalue of S, respectively. We now show how to convert this problem into an SDP.

Recall that S can be factored into $S = QDQ^T$ where Q is an orthonormal matrix (i.e., $Q^{-1} = Q^T$) and D is a diagonal matrix consisting of the eigenvalues of S. The conditions:

$$\lambda I \preceq S \preceq \mu I$$

can be rewritten as:

$$Q(\lambda I)Q^T \preceq QDQ^T \preceq Q(\mu I)Q^T.$$

After premultiplying the above Q^T and postmultiplying Q, these conditions become:

$$\lambda I \preceq D \preceq \mu I$$

which are equivalent to:

$$\lambda \leq \lambda_{\min}(S) \ \text{ and } \ \lambda_{\max}(S) \leq \mu.$$

Therefore EOP can be written as:

$$\begin{array}{lll} EOP: & \underset{w, S, \mu, \lambda}{\text{minimize}} & \mu - \lambda \\ & \text{s.t.} & S = B - \sum_{i=1}^{k} w_i A_i \\ & & \\ & & \lambda I \preceq S \preceq \mu I. \end{array}$$

This last problem is a semidefinite program.

Using constructs such as those shown above, very many other types of eigenvalue optimization problems can be formulated as SDPs.

9 SDP in Control Theory

A variety of control and system problems can be cast and solved as instances of SDP. However, this topic is beyond the scope of these notes.

10 Interior-point Methods for SDP

At the heart of an interior-point method is a barrier function that exerts a repelling force from the boundary of the feasible region. For SDP, we need a barrier function whose values approach $+\infty$ as points X approach the boundary of the semidefinite cone S^n_+.

Let $X \in S^n_+$. Then X will have n eigenvalues, say $\lambda_1(X), \ldots, \lambda_n(X)$ (possibly counting multiplicities). We can characterize the interior of the semidefinite cone as follows:

$$intS^n_+ = \{X \in S^n \mid \lambda_1(X) > 0, \ldots, \lambda_n(X) > 0\}.$$

A natural barrier function to use to repel X from the boundary of S^n_+ then is

$$-\sum_{j=1}^{n} \ln(\lambda_i(X)) = -\ln(\prod_{j=1}^{n} \lambda_i(X)) = -\ln(\det(X)).$$

Consider the logarithmic barrier problem $BSDP(\theta)$ parameterized by the positive barrier parameter θ:

$$
\begin{array}{llll}
BSDP(\theta): & \text{minimize} & C \bullet X - \theta \ln(\det(X)) & \\
& \text{s.t.} & A_i \bullet X = b_i & , i = 1, \ldots, m, \\
& & X \succ 0. &
\end{array}
$$

Let $f_\theta(X)$ denote the objective function of $BSDP(\theta)$. Then it is not too difficult to derive:

$$
\nabla f_\theta(X) = C - \theta X^{-1}, \tag{1}
$$

and so the Karush-Kuhn-Tucker conditions for $BSDP(\theta)$ are:

$$
\begin{cases}
A_i \bullet X = b_i \quad , i = 1, \ldots, m, \\
X \succ 0, \\
C - \theta X^{-1} = \sum_{i=1}^{m} y_i A_i.
\end{cases} \tag{2}
$$

Because X is symmetric, we can factorize X into $X = LL^T$. We then can define

$$
S = \theta X^{-1} = \theta L^{-T} L^{-1},
$$

which implies

$$
\frac{1}{\theta} L^T S L = I,
$$

and we can rewrite the Karush-Kuhn-Tucker conditions as:

$$\left\{\begin{array}{l} A_i \bullet X = b_i \quad, i = 1, \ldots, m, \\ \\ X \succ 0, X = LL^T \\ \\ \sum\limits_{i=1}^{m} y_i A_i + S = C \\ \\ I - \frac{1}{\theta} L^T S L = 0. \end{array}\right. \tag{3}$$

From the equations of (3) it follows that if (X, y, S) is a solution of (3), then X is feasible for SDP, (y, S) is feasible for SDD, and the resulting duality gap is

$$S \bullet X = \sum_{i=1}^{n} \sum_{j=1}^{n} S_{ij} X_{ij} = \sum_{j=1}^{n} (SX)_{jj} = \sum_{j=1}^{n} \theta = n\theta.$$

This suggests that we try solving $BSDP(\theta)$ for a variety of values of θ as $\theta \to 0$.

However, we cannot usually solve (3) exactly, because the fourth equation group is not linear in the variables. We will instead define a "β-approximate solution" of the Karush-Kuhn-Tucker conditions (3). Before doing so, we introduce the following norm on matrices, called the Frobenius norm:

$$\|M\| := \sqrt{M \bullet M} = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} M_{ij}^2} \; .$$

For some important properties of the Frobenius norm, see the last subsection of this section. A "β-approximate solution" of $BSDP(\theta)$ is defined as any solution (X, y, S) of

$$\begin{cases} A_i \bullet X = b_i \quad, i = 1, \ldots, m, \\ \\ X \succ 0, X = LL^T \\ \\ \sum\limits_{i=1}^{m} y_i A_i + S = C \\ \\ \|I - \frac{1}{\theta} L^T S L\| \leq \beta. \end{cases} \tag{4}$$

Lemma 10.1 *If $(\bar{X}, \bar{y}, \bar{S})$ is a β-approximate solution of $BSDP(\theta)$ and $\beta < 1$, then $\bar{X}$ is feasible for SDP, $(\bar{y}, \bar{S})$ is feasible for SDD, and the duality gap satisfies:*

$$n\theta(1 - \beta) \leq C \bullet X - \sum_{i=1}^{m} y_i b_i = \bar{X} \bullet \bar{S} \leq n\theta(1 + \beta). \tag{5}$$

Proof: Primal feasibility is obvious. To prove dual feasibility, we need to show that $\bar{S} \succeq 0$. To see this, define

$$R = I - \frac{1}{\theta} \bar{L}^T \bar{S} \bar{L} \tag{6}$$

and note that $\|R\| \leq \beta < 1$. Rearranging (6), we obtain

$$\bar{S} = \theta \bar{L}^{-T} (I - R) \bar{L}^{-1} \succ 0$$

because $\|R\| < 1$ implies that $I - R \succ 0$. We also have $\bar{X} \bullet \bar{S} = \text{trace}(\bar{X}\bar{S}) = \text{trace}(\bar{L}\bar{L}^T \bar{S}) = \text{trace}(\bar{L}^T \bar{S} \bar{L}) = \theta \text{trace}(I - R) = \theta(n - \text{trace}(R))$. However, $|\text{trace}(R)| \leq \sqrt{n}\|R\| \leq n\beta$, whereby we obtain

$$n\theta(1 - \beta) \leq \bar{X} \bullet \bar{S} \leq n\theta(1 + \beta).$$

q.e.d.

10.1 The Algorithm

Based on the analysis just presented, we are motivated to develop the following algorithm:

Step 0 . Initialization. Data is $(X^0, y^0, S^0, \theta^0)$. $k = 0$. Assume that (X^0, y^0, S^0) is a β-approximate solution of $BSDP(\theta^0)$ for some known value of β that satisfies $\beta < 1$.

Step 1. Set Current values. $(\bar{X}, \bar{y}, \bar{S}) = (X^k, y^k, S^k)$, $\theta = \theta^k$.

Step 2. Shrink θ. Set $\theta^{'} = \alpha\theta$ for some $\alpha \in (0, 1)$. In fact, it will be appropriate to set

$$\alpha = 1 - \frac{\sqrt{\beta} - \beta}{\sqrt{\beta} + \sqrt{n}}$$

Step 3. Compute Newton Direction and Multipliers. Compute the Newton step $D^{'}$ for $BSDP(\theta^{'})$ at $X = \bar{X}$ by factoring $\bar{X} = \bar{L}\bar{L}^T$ and solving the following system of equations in the variables (D, y):

$$\begin{cases} C - \theta^{'}\bar{X}^{-1} + \theta^{'}\bar{X}^{-1}D\bar{X}^{-1} = \sum\limits_{i=1}^{m} y_i A_i \\ \\ A_i \bullet D = 0, \quad i = 1, \ldots, m. \end{cases} \tag{7}$$

Denote the solution to this system by $(D^{'}, y^{'})$.

Step 4. Update All Values.

$$X^{'} = \bar{X} + D^{'}$$

$$S' = C - \sum_{i=1}^{m} y_i' A_i$$

Step 5. Reset Counter and Continue. $(X^{k+1}, y^{k+1}, S^{k+1}) = (X', y', S')$. $\theta^{k+1} = \theta'$. $k \leftarrow k + 1$. Go to Step 1.

Figure 1 shows a picture of the algorithm:

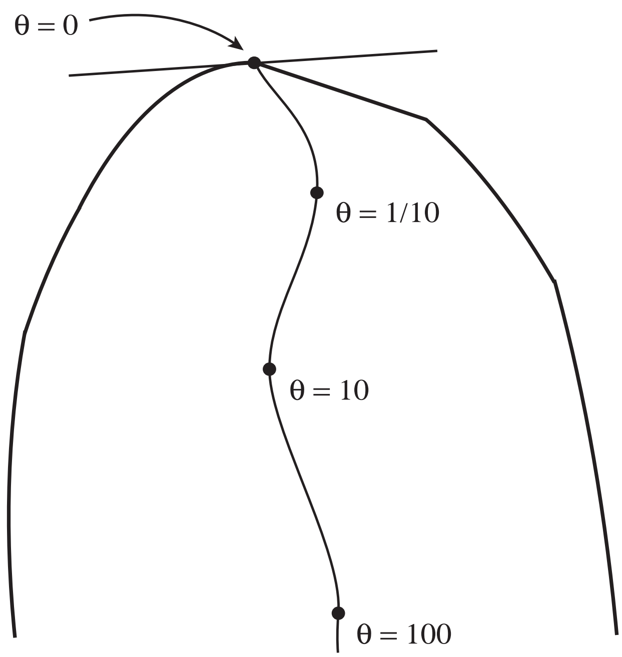

Figure 1: A conceptual picture of the interior-point algorithm.

Some of the unresolved issues regarding this algorithm include:

- how to set the fractional decrease parameter α
- the derivation of the Newton step D' and the multipliers y'

- whether or not successive iterative values (X^k, y^k, S^k) are β-approximate solutions to $BSDP(\theta^k)$, and
- how to get the method started in the first place.

10.2 The Newton Step

Suppose that $\bar{X}$ is a feasible solution to $BSDP(\theta)$:

$$\begin{array}{lll} BSDP(\theta): & \text{minimize} & C \bullet X - \theta \ln(\det(X)) \\ & & \\ & \text{s.t.} & A_i \bullet X = b_i \quad, i = 1, \ldots, m, \\ & & X \succ 0. \end{array}$$

Let us denote the objective function of $BSDP(\theta)$ by $f_\theta(X)$, i.e.,

$$f_\theta(X) = C \bullet X - \theta \ln(\det(X)).$$

Then we can derive:

$$\nabla f_\theta(\bar{X}) = C - \theta \bar{X}^{-1}$$

and the quadratic approximation of $BSDP(\theta)$ at $X = \bar{X}$ can be derived as:

$$\begin{array}{ll} \underset{X}{\text{minimize}} & f_\theta(\bar{X}) + (C - \bar{X}^{-1}) \bullet (X - \bar{X}) + \frac{1}{2}\theta \bar{X}^{-1}(X - \bar{X}) \bullet \bar{X}^{-1}(X - \bar{X}) \\ & \\ \text{s.t.} & A_i \bullet X = b_i, \quad i = 1, \ldots, m. \end{array}$$

Letting $D = X - \bar{X}$, this is equivalent to:

$$\begin{array}{ll} \underset{D}{\text{minimize}} & (C - \theta\bar{X}^{-1}) \bullet D + \frac{1}{2}\theta\bar{X}^{-1}D \bullet \bar{X}^{-1}D \\ \\ \text{s.t.} & A_i \bullet D = 0, \quad i = 1, \ldots, m. \end{array}$$

The solution to this program will be the Newton direction. The Karush-Kuhn-Tucker conditions for this program are necessary and sufficient, and are:

$$\begin{cases} C - \theta\bar{X}^{-1} + \theta\bar{X}^{-1}D\bar{X}^{-1} = \sum_{i=1}^{m} y_i A_i \\ \\ A_i \bullet D = 0, \quad i = 1, \ldots, m. \end{cases} \tag{8}$$

These equations are called the *Normal Equations*. Let $D^{'}$ and $y^{'}$ denote the solution to the Normal Equations. Note in particular from the first equation in (8) that $D^{'}$ must be symmetric. Suppose that $(D^{'}, y^{'})$ is the (unique) solution of the Normal Equations (8). We obtain the new value of the primal variable X by taking the Newton step, i.e.,

$$X^{'} = \bar{X} + D^{'}.$$

We can produce new values of the dual variables (y, S) by setting the new value of y to be $y^{'}$ and by setting $S^{'} = C - \sum_{i=1}^{m} y^{'} A_i$. Using (8), then, we have that

$$S^{'} = \theta\bar{X}^{-1} - \theta\bar{X}^{-1}D^{'}\bar{X}^{-1}. \tag{9}$$

We have the following very powerful convergence theorem which demonstrates the quadratic convergence of Newton's method for this problem, with an explicit guarantee of the range in which quadratic convergence takes place.

Theorem 10.1 (Explicit Quadratic Convergence of Newton's Method). *Suppose that $(\bar{X}, \bar{y}, \bar{S})$ is a β-approximate solution of $BSDP(\theta)$ and $\beta < 1$. Let $(D^{'}, y^{'})$ be the solution to the Normal Equations (8), and let*

$$X^{'} = \bar{X} + D^{'}$$

and

$$S^{'} = \theta \bar{X}^{-1} - \theta \bar{X}^{-1} D^{'} \bar{X}^{-1}.$$

Then $(X^{'}, y^{'}, S^{'})$ is a β^2-approximate solution of $BSDP(\theta)$.

Proof: Our current point $\bar{X}$ satisfies:

$$A_i \bullet \bar{X} = b_i, i = 1, \ldots, m, \bar{X} = \bar{L}\bar{L}^T \succ 0$$

$$\sum_{i=1}^{m} \bar{y}_i A_i + \bar{S} = C$$

$$\|I - \frac{1}{\theta} \bar{L}^T \bar{S} \bar{L}\| \le \beta < 1.$$

Furthermore the Newton direction $D^{'}$ and multipliers $y^{'}$ satisfy:

$$A_i \bullet D^{'} = 0, i = 1, \ldots, m$$

$$\sum_{i=1}^{m} y_i^{'} A_i + S^{'} = C$$

$$X^{'} = \bar{X} + D^{'} = \bar{L}(I + \bar{L}^{-1} D^{'} \bar{L}^{-T}) \bar{L}^T$$

$$S^{'} = \theta \bar{X}^{-1} - \theta \bar{X}^{-1} D^{'} \bar{X}^{-1} = \theta \bar{L}^{-T} (I - \bar{L}^{-1} D^{'} \bar{L}^{-T}) \bar{L}^{-1}.$$

We will first show that $\|\bar{L}^{-1} D^{'} \bar{L}^{-T}\| \le \beta$. It turns out that this is the crucial fact from which everything will follow nicely. To prove this, note that

$$\sum_{i=1}^{m} \bar{y}_i A_i + \bar{S} = C = \sum_{i=1}^{m} y_i^{'} A_i + S^{'} = \sum_{i=1}^{m} y_i^{'} A_i + \theta \bar{L}^{-T} (I - \bar{L}^{-1} D^{'} \bar{L}^{-T}) \bar{L}^{-1}.$$

Taking the inner product with $D^{'}$ yields:

$$\bar{S} \bullet D^{'} = \theta \bar{L}^{-T}\bar{L}^{-1} \bullet D^{'} - \theta \bar{L}^{-T}\bar{L}^{-1}D^{'}\bar{L}^{-T}\bar{L}^{-1} \bullet D^{'},$$

which we can rewrite as:

$$\bar{L}^T\bar{S}\bar{L} \bullet \bar{L}^{-1}D^{'}\bar{L}^{-T} = \theta I \bullet \bar{L}^{-1}D^{'}\bar{L}^{-T} - \theta \bar{L}^{-1}D^{'}\bar{L}^{-T} \bullet \bar{L}^{-1}D^{'}\bar{L}^{-T},$$

which we finally rewrite as:

$$\|\bar{L}^{-1}D^{'}\bar{L}^{-T}\|^2 = \left(I - \frac{1}{\theta}\bar{L}^T\bar{S}\bar{L}\right) \bullet \bar{L}^{-1}D^{'}\bar{L}^{-T}.$$

Invoking the Cauchy-Schwartz inequality we obtain:

$$\|\bar{L}^{-1}D^{'}\bar{L}^{-T}\|^2 \leq \|I - \frac{1}{\theta}\bar{L}^T\bar{S}\bar{L}\|\|\bar{L}^{-1}D^{'}\bar{L}^{-T}\| \leq \beta\|\bar{L}^{-1}D^{'}\bar{L}^{-T}\|,$$

from which we see that $\|\bar{L}^{-1}D^{'}\bar{L}^{-T}\| \leq \beta$.

It therefore follows that

$$X^{'} = \bar{L}(I + \bar{L}^{-1}D^{'}\bar{L}^{-T})\bar{L}^T \succ 0$$

and

$$S^{'} = \theta \bar{L}^{-T}(I - \bar{L}^{-1}D^{'}\bar{L}^{-T})\bar{L}^{-1} \succ 0,$$

since $\|\bar{L}^{-1}D^{'}\bar{L}^{-T}\| \leq \beta < 1$, which guarantees that $I \pm \bar{L}^{-1}D^{'}\bar{L}^{-T} \succ 0$.

Next, factorize

$$I + \bar{L}^{-1}D^{'}\bar{L}^{-T} = M^2,$$

(where $M = M^T$) and note that

$$X^{'} = \bar{L}MM\bar{L}^T = L^{'}(L^{'})^T$$

where we define $L^{'} = \bar{L}M$. Then note that

$$I - \frac{1}{\theta}(L^{'})^T S^{\prime} L^{'} = I - \frac{1}{\theta}M\bar{L}^T S^{'}\bar{L}M$$

$$= I - \frac{1}{\theta}M\bar{L}^T(\theta\bar{L}^{-T}(I - \bar{L}^{-1}D^{'}\bar{L}^{-T})\bar{L}^{-1})\bar{L}M = I - M(I - \bar{L}^{-1}D^{'}\bar{L}^{-T})M$$

$$= I - MM + M(\bar{L}^{-1}D^{'}\bar{L}^{-T})M = I - MM + M(MM - I)M = (I - MM)(I - MM)$$
$$= (\bar{L}^{-1}D^{'}\bar{L}^{-T})(\bar{L}^{-1}D^{'}\bar{L}^{-T}).$$

From this we next have:

$$\|I - \frac{1}{\theta}(L^{'})^T S^{'} L^{'}\| = \|(\bar{L}^{-1}D^{'}\bar{L}^{-T})(\bar{L}^{-1}D^{'}\bar{L}^{-T})\| \leq \|(\bar{L}^{-1}D^{'}\bar{L}^{-T})\|^2 \leq \beta^2.$$

This shows that $(X^{'}, y^{'}, S^{'})$ is a β^2-approximate solution of $BSDP(\theta)$.
q.e.d.

10.3 Complexity Analysis of the Algorithm

Theorem 10.2 (Relaxation Theorem). *Suppose that $(\bar{X}, \bar{y}, \bar{S})$ is a β-approximate solution of $BSDP(\theta)$ and $\beta < 1$. Let*

$$\alpha = 1 - \frac{\sqrt{\beta} - \beta}{\sqrt{\beta} + \sqrt{n}}$$

and let $\theta^{'} = \alpha\theta$. Then $(\bar{X}, \bar{y}, \bar{S})$ is a $\sqrt{\beta}$-approximate solution of $BSDP(\theta^{'})$.

Proof: The triplet $(\bar{X}, \bar{y}, \bar{S})$ satisfies $A_i \bullet \bar{X} = b_i, i = 1, \ldots, m, \bar{X} \succ 0$, and $\sum_{i=1}^{m} \bar{y}_i A_i + \bar{S} = C$, and so it remains to show that

$$\left\| \frac{1}{\theta^{'}} \bar{L}^T \bar{S} \bar{L} - I \right\| \leq \sqrt{\beta},$$

where $\bar{X} = \bar{L}\bar{L}^T$. We have

$$\left\| \frac{1}{\theta^{'}} \bar{L}^T \bar{S} \bar{L} - I \right\| = \left\| \frac{1}{\alpha\theta} \bar{L}^T \bar{S} \bar{L} - I \right\| = \left\| \frac{1}{\alpha} \left(\frac{1}{\theta} \bar{L}^T \bar{S} \bar{L} - I \right) - \left(1 - \frac{1}{\alpha} \right) I \right\|$$
$$\leq \left(\frac{1}{\alpha} \right) \left\| \frac{1}{\theta} \bar{L}^T \bar{S} \bar{L} - I \right\| + \left| \frac{1 - \alpha}{\alpha} \right| \|I\|$$
$$\leq \frac{\beta}{\alpha} + \left(\frac{1 - \alpha}{\alpha} \right) \sqrt{n} \quad = \quad \frac{\beta + \sqrt{n}}{\alpha} - \sqrt{n}$$
$$= \sqrt{\beta} + \sqrt{n} - \sqrt{n} = \sqrt{\beta}.$$

q.e.d.

Theorem 10.3 (Convergence Theorem). *Suppose that (X^0, y^0, S^0) is a β-approximate solution of $BSDP(\theta^0)$ and $\beta < 1$. Then for all $k = 1, 2, 3, ...,$ (X^k, y^k, S^k) is a β-approximate solution of $BSDP(\theta^k)$.*

Proof: By induction, suppose that the theorem is true for iterates $0, 1, 2, ..., k$.

Then (X^k, y^k, S^k) is a β-approximate solution of $BSDP(\theta^k)$.

From the Relaxation Theorem, (X^k, y^k, S^k) is a $\sqrt{\beta}$-approximate solution of $BSDP(\theta^{k+1})$ where $\theta^{k+1} = \alpha\theta^k$.

From the Quadratic Convergence Theorem, $(X^{k+1}, y^{k+1}, S^{k+1})$ is a β-approximate solution of $BSDP(\theta^{k+1})$.

Therefore, by induction, the theorem is true for all values of k.
q.e.d.

Figure 2 shows a better picture of the algorithm:

Theorem 10.4 (Complexity Theorem). *Suppose that (X^0, y^0, S^0) is a $\beta = \frac{1}{4}$-approximate solution of $BSDP(\theta^0)$. In order to obtain primal and dual feasible solutions (X^k, y^k, S^k) with a duality gap of at most ϵ, one needs to run the algorithm for at most*

$$k = \left\lceil 6\sqrt{n} \ln\left(\frac{1.25}{0.75} \frac{X^0 \bullet S^0}{\epsilon}\right)\right\rceil$$

iterations.

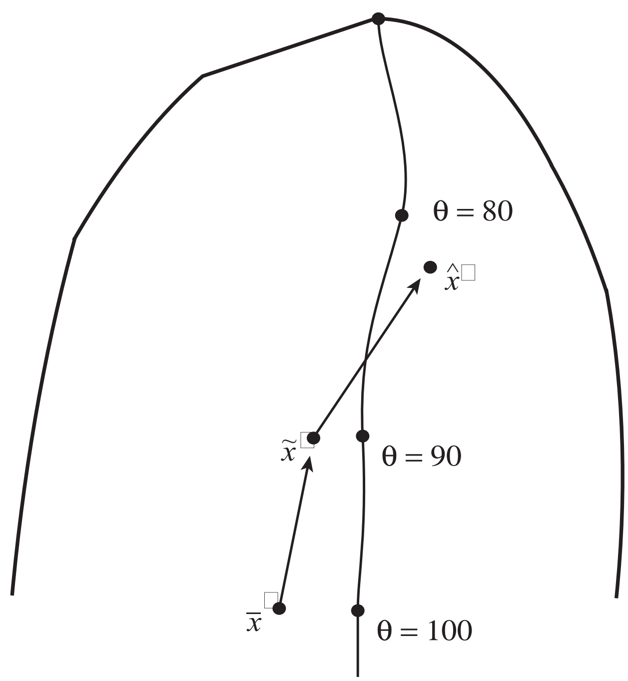

Figure 2: Another picture of the interior-point algorithm.

Proof: Let k be as defined above. Note that

$$\alpha = 1 - \frac{\sqrt{\beta}-\beta}{\sqrt{\beta}+\sqrt{n}} = 1 - \frac{\frac{1}{2}-\frac{1}{4}}{\left(\frac{1}{2}+\sqrt{n}\right)} = 1 - \frac{1}{2+4\sqrt{n}} \leq 1 - \frac{1}{6\sqrt{n}}.$$

Therefore

$$\theta^k \leq \left(1 - \frac{1}{6\sqrt{n}}\right)^k \theta^0.$$

This implies that

$$C \bullet X^k - \sum_{i=1}^m b_i y_i^k = X^k \bullet S^k \leq \theta^k n(1+\beta) \leq \left(1 - \frac{1}{6\sqrt{n}}\right)^k (1.25n\theta^0)$$

$$\leq \left(1 - \frac{1}{6\sqrt{n}}\right)^k (1.25n) \left(\frac{X^0 \bullet S^0}{0.75n}\right),$$

from (5). Taking logarithms, we obtain

$$\ln\left(C \bullet X^k - \sum_{i=1}^m b_i y_i^k\right) \leq k \ln\left(1 - \frac{1}{6\sqrt{n}}\right) + \ln\left(\frac{1.25}{0.75} X^0 \bullet S^0\right)$$

$$\leq \frac{-k}{6\sqrt{n}} + \ln\left(\frac{1.25}{0.75} X^0 \bullet S^0\right)$$

$$\leq -\ln\left(\frac{1.25}{0.75}\frac{X^0 \bullet S^0}{\epsilon}\right) + \ln\left(\frac{1.25}{0.75} X^0 \bullet S^0\right) = \ln(\epsilon).$$

Therefore $C \bullet X^k - \sum_{i=1}^m b_i y_i^k \leq \epsilon$.

q.e.d.

10.4 How to Start the Method from a Strictly Feasible Point

The algorithm and its performance relies on having a starting point (X^0, y^0, S^0) that is a β-approximate solution of the problem $BSDP(\theta^0)$. In this subsection, we show how to obtain such a starting point, given a positive definite feasible solution X^0 of SDP.

We suppose that we are given a target value θ^0 of the barrier parameter, and we are given $X = X^0$ that is feasible for $BSDP(\theta^0)$, that is, $A_i \bullet X^0 = b_i, i = 1, \ldots, m$, and $X^0 \succ 0$. We will attempt to approximately solve $BSDP(\theta^0)$ starting at $X = X^0$, using the Newton direction at each iteration. The formal statement of the algorithm is as follows:

Step 0 . Initialization. Data is (X^0, θ^0). $k = 0$. Assume that X^0 satisfies $A_i \bullet X^0 = b_i, i = 1, \ldots, m, X^0 \succ 0$.

Step 1. Set Current values. $\bar{X} = X^k$. Factor $\bar{X} = \bar{L}\bar{L}^T$.

Step 2. Compute Newton Direction and Multipliers. Compute the Newton step $D^{'}$ for $BSDP(\theta^0)$ at $X = \bar{X}$ by solving the following system of equations in the variables (D, y):

$$\left\{ \begin{array}{l} C - \theta^0 \bar{X}^{-1} + \theta^0 \bar{X}^{-1} D \bar{X}^{-1} = \sum\limits_{i=1}^{m} y_i A_i \\ \\ A_i \bullet D = 0, \quad i = 1, \ldots, m. \end{array} \right. \tag{10}$$

Denote the solution to this system by $(D^{'}, y^{'})$. Set

$$S^{'} = C - \sum_{i=1}^{m} y_i^{'} A_i.$$

Step 3. Test the Current Point. If $\|\bar{L}^{-1} D^{'} \bar{L}^{-T}\| \leq \frac{1}{4}$, stop. In this case, $\bar{X}$ is a $\frac{1}{4}$-approximate solution of $BSDP(\theta^0)$, along with the dual values $(y^{'}, S^{'})$.

Step 4. Update Primal Point.

$$X^{'} = \bar{X} + \alpha D^{'}$$

where
$$\alpha = \frac{0.2}{\|\bar{L}^{-1}D'\bar{L}^{-T}\|}.$$
Alternatively, α can be computed by a line-search of $f_{\theta^0}(\bar{X} + \alpha D^{'})$.

Step 5. Reset Counter and Continue. $X^{k+1} \leftarrow X^{'}$, $k \leftarrow k + 1$. Go to Step 1.

The following proposition validates Step 3 of the algorithm:

Proposition 10.1 *Suppose that $(D^{'}, y^{'})$ is the solution of the Normal equations (10) for the point $\bar{X}$ for the given value θ^0 of the barrier parameter, and that*
$$\|\bar{L}^{-1}D^{'}\bar{L}^{-T}\| \leq \frac{1}{4}.$$
Then $\bar{X}$ is a $\frac{1}{4}$-approximate solution of $BSDP(\theta^0)$.

Proof: We must exhibit values (y, S) that satisfy $\sum_{i=1}^{m} y_i A_i + S = C$ and
$$\left\| I - \frac{1}{\theta^0}\bar{L}^T S \bar{L} \right\| \leq \frac{1}{4}.$$
Let $(D^{'}, y^{'})$ solve the Normal equations (10), and let $S^{'} = C - \sum_{i=1}^{m} y_i^{'} A_i$. Then we have from (10) that
$$I - \frac{1}{\theta^0}\bar{L}^T S^{'} \bar{L} = I - \frac{1}{\theta^0}\bar{L}^T(\theta^0(\bar{L}^{-T}\bar{L}^{-1} - \bar{L}^{-T}\bar{L}^{-1}D^{'}\bar{L}^{-T}\bar{L}^{-1}))\bar{L} = \bar{L}^{-1}D^{'}\bar{L}^{-T},$$
whereby
$$\|I - \frac{1}{\theta^0}\bar{L}^T S^{'} \bar{L}\| = \|\bar{L}^{-1}D^{'}\bar{L}^{-T}\| \leq \frac{1}{4}.$$

q.e.d.

The next proposition shows that whenever the algorithm proceeds to Step 4, then the objective function $f_{\theta^0}(X)$ decreases by at least $0.025\theta^0$:

Proposition 10.2 *Suppose that $\bar{X}$ satisfies $A_i \bullet \bar{X} = b_i, i = 1, \ldots, m$, and $\bar{X} \succ 0$. Suppose that (D', y') is the solution of the Normal equations (10) for the point $\bar{X}$ for a given value θ^0 of the barrier parameter, and that*

$$\|\bar{L}^{-1} D' \bar{L}^{-T}\| > \frac{1}{4}.$$

Then for all $\gamma \in [0, 1)$,

$$f_{\theta^0}\left(\bar{X} + \frac{\gamma}{\|\bar{L}^{-1} D' \bar{L}^{-T}\|} D'\right) \le f_{\theta^0}(\bar{X}) + \theta^0 \left(-\gamma \|\bar{L}^{-1} D' \bar{L}^{-T}\| + \frac{\gamma^2}{2(1-\gamma)}\right).$$

In particular,

$$f_{\theta^0}\left(\bar{X} + \frac{0.2}{\|\bar{L}^{-1} D' \bar{L}^{-T}\|} D'\right) \le f_{\theta^0}(\bar{X}) - 0.025\theta^0. \tag{11}$$

In order to prove this proposition, we will need two powerful facts about the logarithm function:

Fact 1. Suppose that $|x| \le \delta < 1$. Then

$$\ln(1 + x) \ge x - \frac{x^2}{2(1-\delta)}.$$

Proof: We have:

$$\begin{aligned}
\ln(1+x) &= x - \tfrac{x^2}{2} + \tfrac{x^3}{3} - \tfrac{x^4}{4} + \ldots \\
&\ge x - \tfrac{|x|^2}{2} - \tfrac{|x|^3}{3} - \tfrac{|x|^4}{4} - \ldots \\
&\ge x - \tfrac{|x|^2}{2} - \tfrac{|x|^3}{2} - \tfrac{|x|^4}{2} - \ldots \\
&= x - \tfrac{x^2}{2}\left(1 + |x| + |x|^2 + |x|^3 + \ldots\right) \\
&= x - \tfrac{x^2}{2(1-|x|)} \\
&\ge x - \tfrac{x^2}{2(1-\delta)}.
\end{aligned}$$

q.e.d.

Fact 2. Suppose that $R \in S^n$ and that $\|R\| \leq \gamma < 1$. Then

$$\ln(\det(I+R)) \geq I \bullet R - \frac{\gamma^2}{2(1-\gamma)}.$$

Proof: Factor $R = QDQ^T$ where Q is orthonormal and D is a diagonal matrix of the eigenvalues of R. Then first note that $\|R\| = \sqrt{\sum_{j=1}^{n} D_{jj}^2}$. We then have:

$$\begin{array}{rcl} \ln(\det(I+R)) & = & \ln(\det(I+QDQ^T)) \\ & = & \ln(\det(I+D)) \\ & = & \sum_{j=1}^{n} \ln(1+D_{jj}) \\ & \geq & \sum_{j=1}^{n} \left(D_{jj} - \frac{D_{jj}^2}{2(1-\gamma)} \right) \\ & = & I \bullet D - \frac{\|R\|^2}{2(1-\gamma)} \\ & \geq & I \bullet Q^T R Q - \frac{\gamma^2}{2(1-\gamma)} \\ & = & I \bullet R - \frac{\gamma^2}{2(1-\gamma)} \end{array}$$

q.e.d.

Proof of Proposition 10.2. Let

$$\alpha = \frac{\gamma}{\|\bar{L}^{-1} D' \bar{L}^{-T}\|},$$

and notice that

$$\|\alpha \bar{L}^{-1} D' \bar{L}^{-T}\| = \gamma.$$

Then

$$
\begin{aligned}
f_{\theta^0}\left(\bar{X} + \tfrac{\gamma}{\|\bar{L}^{-1}D'\bar{L}^{-T}\|}D'\right) &= f_{\theta^0}\left(\bar{X} + \alpha D'\right) \\
&= C \bullet \bar{X} + \alpha C \bullet D' - \theta^0 \ln(\det(\bar{L}(I + \alpha \bar{L}^{-1} D' \bar{L}^{-T})\bar{L}^T)) \\
&= C \bullet \bar{X} - \theta^0 \ln(\det(\bar{X})) + \alpha C \bullet D' - \theta^0 \ln(\det(I + \alpha \bar{L}^{-1} D' \bar{L}^{-T})) \\
&\leq f_{\theta^0}(\bar{X}) + \alpha C \bullet D' - \theta^0 \alpha I \bullet \bar{L}^{-1} D' \bar{L}^{-T} + \theta^0 \tfrac{\gamma^2}{2(1-\gamma)} \\
&= f_{\theta^0}(\bar{X}) + \alpha C \bullet D' - \theta^0 \alpha \bar{L}^{-T} \bar{L}^{-1} \bullet D' + \theta^0 \tfrac{\gamma^2}{2(1-\gamma)} \\
&= f_{\theta^0}(\bar{X}) + \alpha \left(C - \theta^0 \bar{X}^{-1}\right) \bullet D' + \theta^0 \tfrac{\gamma^2}{2(1-\gamma)}.
\end{aligned}
$$

Now, (D', y') solve the Normal equations:

$$
\begin{cases}
C - \theta^0 \bar{X}^{-1} + \theta^0 \bar{X}^{-1} D' \bar{X}^{-1} = \sum\limits_{i=1}^{m} y_i' A_i \\
\\
A_i \bullet D' = 0, \quad i = 1, \ldots, m.
\end{cases}
\tag{12}
$$

Taking the inner product of both sides of the first equation above with D' and rearranging yields:

$$
\theta^0 \bar{X}^{-1} D' \bullet \bar{X}^{-1} D' = -(C - \theta^0 \bar{X}^{-1}) \bullet D'.
$$

Substituting this in our inequality above yields:

$$
\begin{aligned}
f_{\theta^0}\left(\bar{X} + \frac{\gamma}{\|\bar{L}^{-1}D'\bar{L}^{-T}\|}D'\right) &\leq f_{\theta^0}(\bar{X}) - \alpha\theta^0\bar{X}^{-1}D' \bullet \bar{X}^{-1}D' + \theta^0\frac{\gamma^2}{2(1-\gamma)} \\
&= f_{\theta^0}(\bar{X}) - \alpha\theta^0\bar{L}^{-1}D'\bar{L}^{-T} \bullet \bar{L}^{-1}D'\bar{L}^{-T} + \theta^0\frac{\gamma^2}{2(1-\gamma)} \\
&= f_{\theta^0}(\bar{X}) - \alpha\theta^0\|\bar{L}^{-1}D'\bar{L}^{-T}\|^2 + \theta^0\frac{\gamma^2}{2(1-\gamma)} \\
&= f_{\theta^0}(\bar{X}) - \theta^0\gamma\|\bar{L}^{-1}D'\bar{L}^{-T}\| + \theta^0\frac{\gamma^2}{2(1-\gamma)} \\
&= f_{\theta^0}(\bar{X}) + \theta^0\left(-\gamma\|\bar{L}^{-1}D'\bar{L}^{-T}\| + \frac{\gamma^2}{2(1-\gamma)}\right).
\end{aligned}
$$

Subsituting $\gamma = 0.2$ and and $\|\bar{L}^{-1}D'\bar{L}^{-T}\| > \frac{1}{4}$ yields the final result.
q.e.d.

Last of all, we prove a bound on the number of iterations that the algorithm will need in order to find a $\frac{1}{4}$-approximate solution of $BSDP(\theta^0)$:

Proposition 10.3 *Suppose that X^0 satisfies $A_i \bullet X^0 = b_i, i = 1, \ldots, m$, and $X^0 \succ 0$. Let θ^0 be given and let $f^*_{\theta^0}$ be the optimal objective function value of $BSDP(\theta^0)$. Then the algorithm initiated at X^0 will find a $\frac{1}{4}$-approximate solution of $BSDP(\theta^0)$ in at most*

$$
k = \left\lceil \frac{f_{\theta^0}(X^0) - f^*_{\theta^0}}{0.025\theta^0} \right\rceil
$$

iterations.

Proof: This follows immediately from (11). Each iteration that is not a $\frac{1}{4}$-approximate solution decreases the objective function $f_{\theta^0}(X)$ of $BSDP(\theta^0)$ by at least $0.025\theta^0$. Therefore, there cannot be more than

$$
\left\lceil \frac{f_{\theta^0}(X^0) - f^*_{\theta^0}}{0.025\theta^0} \right\rceil
$$

iterations that are not $\frac{1}{4}$-approximate solutions of $BSDP(\theta^0)$.
q.e.d.

10.5 Some Properties of the Frobenius Norm

Proposition 10.4 *If $M \in S^n$, then*

1. $\|M\| = \sqrt{\sum_{j=1}^{n} (\lambda_j(M))^2}$, *where $\lambda_1(M), \lambda_2(M), \ldots, \lambda_n(M)$ is an enumeration of the n eigenvalues of M.*
2. *If λ is any eigenvalue of M, then $|\lambda| \le \|M\|$.*
3. $|trace(M)| \le \sqrt{n}\|M\|$.
4. *If $\|M\| < 1$, then $I + M \succ 0$.*

Proof: We can factorize $M = QDQ^T$ where Q is orthonormal and D is a diagonal matrix of the eigenvalues of M. Then

$$\|M\| = \sqrt{M \bullet M} = \sqrt{QDQ^T \bullet QDQ^T} = \sqrt{\text{trace}(QDQ^TQDQ^T)}$$

$$= \sqrt{\text{trace}(Q^TQDQ^TQD)} = \sqrt{\text{trace}(DD)} = \sqrt{\sum_{j=1}^{n} (\lambda_j(M))^2}.$$

This proves the first two assertions. To prove the third assertion, note that

$$\text{trace}(M) = \text{trace}(QDQ^T) = \text{trace}(Q^TQD)$$

$$= \text{trace}(D) = \sum_{j=1}^{n} \lambda_j(M) \le \sqrt{n}\sqrt{\sum_{j=1}^{n} (\lambda_j(M))^2} = \sqrt{n}\|M\|.$$

To prove the fourth assertion, let $\lambda^{'}$ be an eigenvalue of $I + M$. Then $\lambda^{'} = 1 + \lambda$ where λ is an eigenvalue of M. However, from the second assertion, $\lambda^{'} = 1 + \lambda \ge 1 - \|M\| > 0$, and so $M \succ 0$.
q.e.d.

Proposition 10.5 *If $A, B \in S^n$, then $\|AB\| \leq \|A\|\|B\|$.*

Proof: We have

$$\|AB\| = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\sum_{k=1}^{n} A_{ik}B_{kj}\right)^2}$$

$$\leq \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\left(\sum_{k=1}^{n} A_{ik}^2\right)\left(\sum_{k=1}^{n} B_{kj}^2\right)\right)}$$

$$= \sqrt{\left(\sum_{i=1}^{n}\sum_{k=1}^{n} A_{ik}^2\right)\left(\sum_{j=1}^{n}\sum_{k=1}^{n} B_{kj}^2\right)} = \|A\|\|B\|.$$

q.e.d.

11 Issues in Solving SDP using the Ellipsoid Algorithm

To see how the ellipsoid algorithm is used to solve a semidefinite program, assume for convenience that the format of the problem is that of the dual problem SDD. Then the feasible region of the problem can be written as:

$$F = \left\{(y_1, \ldots, y_m) \in \Re^m \mid C - \sum_{i=1}^{m} y_i A_i \succeq 0\right\},$$

and the objective function is $\sum_{i=1}^{m} y_i b_i$. Note that F is just a convex region in $\Re^m$.

Recall that at any iteration of the ellipsoid algorithm, the set of solutions of SDD is known to lie in the current ellipsoid, and the center of the current ellipsoid is, say, $\bar{y} = (\bar{y}_1, \ldots, \bar{y}_m)$. If $\bar{y} \in F$, then we perform an optimality cut of the form $\sum_{i=1}^m y_i b_i \geq \sum_{i=1}^m \bar{y}_i b_i$, and use standard formulas to update the ellipsoid and its new center. If $\bar{y} \notin F$, then we perform a feasibility cut by computing a vector $\bar{h} \in \Re^m$ such that $\bar{h}^T y > \bar{h}^T \bar{y}$ for all $y \in F$.

There are four issues that must be resolved in order to implement the above version of the ellipsoid algorithm to solve SDD:

1. Testing if $\bar{y} \in F$. This is done by computing the matrix $\bar{S} = C - \sum_{i=1}^m \bar{y}_i A_i$. If $\bar{S} \succeq 0$, then $\bar{y} \in F$. Testing if the matrix $\bar{S} \succeq 0$ can be done by computing an exact Cholesky factorization of $\bar{S}$, which takes $O(n^3)$ operations, assuming that computing square roots can be performed (exactly) in one operation.

2. Computing a feasibility cut. As above, testing if $\bar{S} \succeq 0$ can be computed in $O(n^3)$ operations. If $\bar{S} \not\succeq 0$, then again assuming exact arithmetic, we can find an n-vector $\bar{v}$ such that $\bar{v}^T \bar{S} \bar{v} < 0$ in $O(n^3)$ operations as well. Then the feasiblity cut vector $\bar{h}$ is computed by the formula:

$$\bar{h}_i = \bar{v}^T A_i \bar{v}, \quad i = 1, \ldots, m,$$

   whose computation requires $O(mn^2)$ operations. Notice that for any $y \in F$, that

$$\begin{aligned} y^T \bar{h} &= \sum_{i=1}^m y_i \bar{v}^T A_i \bar{v} = \bar{v}^T \left(\sum_{i=1}^m y_i A_i \right) \bar{v} \\ &\leq \bar{v}^T C \bar{v} < \bar{v}^T \left(\sum_{i=1}^m \bar{y}_i A_i \right) \bar{v} = \sum_{i=1}^m \bar{y}_i \bar{v}^T A_i \bar{v} = \bar{y}^T \bar{h}, \end{aligned}$$

   thereby showing $\bar{h}$ indeed provides a feasibility cut for F at $y = \bar{y}$.

3. Starting the ellipsoid algorithm. We need to determine an upper bound R on the distance of some optimal solution y^* from the origin. This

cannot be done by examining the input length of the data, as is the case in linear programming. One needs to know some special information about the specific problem at hand in order to determine R before solving the semidefinite program.

4. Stopping the ellipsoid algorithm. Suppose that we seek an ϵ-optimal solution of SDD. In order to prove a complexity bound on the number of iterations needed to find an ϵ-optimal solution, we need to know beforehand the radius r of a Euclidean ball that is contained in the set of ϵ-optimal solutions of SDD. The value of r also *cannot* be determined by examining the input length of the data, as is the case in linear programming. One needs to know some special information about the specific problem at hand in order to determine r before solving the semidefinite program.

12 Current Research in SDP

There are many very active research areas in semidefinite programming in nonlinear (convex) programming, in combinatorial optimization, and in control theory. In the area of convex analysis, recent research topics include the geometry and the boundary structure of SDP feasible regions (including notions of degeneracy) and research related to the computational complexity of SDP such as decidability questions, certificates of infeasibility, and duality theory. In the area of combinatorial optimization, there has been much research on the practical and the theoretical use of SDP relaxations of hard combinatorial optimization problems. As regards interior point methods, there are a host of research issues, mostly involving the development of different interior point algorithms and their properties, including rates of convergence, performance guarantees, etc.

13 Computational State of the Art of SDP

Because SDP has so many applications, and because interior point methods show so much promise, perhaps the most exciting area of research on SDP has to do with computation and implementation of interior point algorithms for solving SDP. Much research has focused on the practical efficiency of interior point methods for SDP. However, in the research to date, computational issues have arisen that are much more complex than those for linear programming, and these computational issues are only beginning to be well-understood. They probably stem from a variety of factors, including the fact that SDP is not guaranteed to have strictly complementary optimal solutions (as is the case in linear programming). Finally, because SDP is such a new field, there is no representative suite of practical problems on which to test algorithms, i.e., there is no equivalent version of the `netlib` suite of industrial linear programming problems.

A good website for semidefinite programming is:

`http://www.zib.de/helmberg/semidef.html`.

14 Exercises

1. For a (square) matrix $M \in \mathbb{R}^{n\times n}$, define $\text{trace}(M) = \sum_{j=1}^{n} M_{jj}$, and for two matrices $A, B \in \mathbb{R}^{k\times l}$ define

$$A \bullet B := \sum_{i=1}^{k} \sum_{j=1}^{l} A_{ij} B_{ij} \ .$$

   Prove that:

   (a) $A \bullet B = \text{trace}(A^T B)$.

   (b) $\text{trace}(MN) = \text{trace}(NM)$.

2. Let $S_+^{k\times k}$ denote the cone of positive semi-definite symmetric matrices, namely $S_+^{k\times k} = \{X \in S^{k\times k} \mid v^T X v \geq 0 \text{ for all } v \in \Re^n\}$. Considering $S_+^{k\times k}$ as a cone, prove that $\left(S_+^{k\times k}\right)^* = S_+^{n\times n}$, thus showing that $S_+^{k\times k}$ is self-dual.

3. Consider the problem:

$$\text{(P)}: \quad \text{minimize}_d \quad d^T Q d$$
$$\text{s.t.} \quad d^T M d = 1 \ .$$

If $M \succ 0$, show that (P) is equivalent to:

$$\text{(S)}: \quad \text{minimize}_X \quad Q \bullet X$$
$$\text{s.t.} \quad M \bullet X = 1$$
$$X \succeq 0 \ .$$

What is the SDP dual of (S)?

4. Prove that

$$X \succeq x x^T$$

if and only if

$$\begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix} \succeq 0 \ .$$

5. Let $K := \{X \in S^{k \times k} \mid X \succeq 0\}$ and define

$$K^* := \{S \in S^{k \times k} \mid S \bullet X \geq 0 \text{ for all } X \succeq 0\} \ .$$

Prove that $K^* = K$.

6. Let $\lambda_{\min}$ denote the smallest eigenvalue of the symmetric matrix Q. Show that the following three optimization problems each have optimal objective function value equal to $\lambda_{\min}$.

$$\text{(P1)}: \quad \text{minimize}_d \quad d^T Q d$$
$$\text{s.t.} \quad d^T I d = 1 \ .$$

$$\text{(P2)}: \quad \text{maximize}_\lambda \quad \lambda$$
$$\text{s.t.} \quad Q \succeq \lambda I \ .$$

$$\begin{array}{lll}
(\mathrm{P3}): & \mathrm{minimize}_X & Q \bullet X \\
& & \\
& \mathrm{s.t.} & I \bullet X = 1 \\
& & X \succeq 0 \ .
\end{array}$$

7. Suppose that $Q \succeq 0$. Prove the following:

$$x^T Q x + q^T x + c \leq 0$$

if and only if there exists W for which

$$\begin{pmatrix} c & \frac{1}{2}q^T \\ \frac{1}{2}q & Q \end{pmatrix} \bullet \begin{pmatrix} 1 & x^T \\ x & W \end{pmatrix} \leq 0 \ \text{ and } \ \begin{pmatrix} 1 & x^T \\ x & W \end{pmatrix} \succeq 0 \ .$$

Hint: use the property shown in Exercise 4.

8. Consider the matrix

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

where A, B are symmetric matrices and A is nonsingular. Prove that $M \succeq 0$ if and only if $C - B^T A^{-1} B \succeq 0$.