King Fahd University of Petroleum & Minerals

DEPARTMENT OF MATHEMATICAL SCIENCES

Technical Report Series

TR290

April 2003

New Formulations of the Hankel Matrix Approximation Problem

M.M. Alshahrani, S. Al-Homidan

DHAHRAN 31261 ● SAUDI ARABIA ● www.kfupm.edu.sa/math/ ● E-mail: mathdept@kfupm.edu.sa

New formulations of the Hankel Matrix Approximation Problem

M. M. Alshahrani[1] S. Al-Homidan[2]

Abstract

The problem of finding the closest positive semidefinite Hankel matrix to a given data covariance matrix, computed from a data sequence is considered. It will be solved using the modified alternating projection method. New reformulations of the problem will be proposed in the form of a semidefinite programming problem and then in the form of a mixed semidefinite and second-order cone optimization problem.

[1]Department of Mathematics, Dammam Teachers' College, P.O. Box 14262, Dammam 31424, SAUDI ARABIA **mmogib@awalnet.net.sa**

[2]Department of Mathematical Sciences, King Fahad University of Petroleum and Minerals, KFUPM Box 119, Dhahran 31261, SAUDI ARABIA **homidan@kfupm.edu.sa**

1 Introduction

In some application areas, such as digital signal processing and control theory, it is required to compute the closest, in some sense, positive semidefinite Hankel matrix, with no restriction on its rank, to a given data covariance matrix, computed from a data sequence. This problem was studied by Macinnes [16]. Similar problems involving structured covariance estimation were discussed in [13, 11, 22]. Related problems occur in many engineering and statistics applications [8].

The problem was formulated as a nonlinear minimization problem with positive semidefinite Hankel matrix as constraints in [2] and then was solved by l_2 Sequential Quadratic Programming (l_2SQP) method. Another approach to deal with this problem was to solve it as a smooth unconstrained minimization problem [1]. Other methods to solve this problem or similar problems can be found in [16, 11, 13].

Our work is mainly casting the problem: first as a semidefinite programming problem and second as a mixed semidefinite and second-order cone optimization problem. A semidefinite programming (SDP) problem is to minimize a linear objective function subject to constraints over the cone of positive semidefinite matrices. It is a relatively new field of mathematical programming, and most of the papers on SDP were written in 1990s, although its roots can be traced back to a few decades earlier (see Bellman and Fan [6]). SDP problems are of great interest due to many reasons , e.g., SDP contains important classes of problems as special cases, such as linear and quadratic programming. Applications of SDP exist in combinatorial optimization, approximation theory, system and control theory, and mechanical and electrical engineering. SDP problems can be solved very efficiently in polynomial time by interior point algorithms [26, 28, 9, 5, 18].

The constraints in a mixed semidefinite and second-order cone optimization problem are constraints over the positive semidefinite and the second-order cones. Although the second-order cone constraints can be seen as positive semidefinite constraints, recent research has shown that it is more effecient to deal with mixed problems rather than the semidefinite programming problem. Nestrov et. al. [18] can be considered as the first paper to deal with mixed semidefinite and second-order cone optimization problems. However, the area was really brought to life by Alizadeh et al. [4] with the introduction of SDPPack, a software package for solving optimization problems from this class. The practical importance of second-order programming was demonstrated by Lobo et al. [15] and many subsequent papers. The interior point methods were recently extended to deal with mixed problems [20]. One class of these interior point methods is the primal-dual path-following methods. These methods are considered the most successful interior point algorithms for linear programming. Their extension from linear to semidefinite and then mixed problems has followed the same trends. One of the most successful implementation of primal-dual path-following methods is in the software SDPT3 by Toh et al. [25, 23].

Similar problems, such as the problem of minimizing the spectral norm of a matrix was first formulated as a semidefinite programming problem in [26, 24]. Then, these problems and some others were formulated as a mixed semidefinite and second-order cone optimization problems [15, 3, 21]. None of these formulations exploited the special structure our problem has.

Before we go any further, we should introduce some notations. Throughout this paper, we will denote the set of all $n \times n$ real symmetric matrices by $\mathcal{S}_n$, the cone of the $n \times n$ real symmetric positive semidefinite matrices by $\mathcal{S}_n^+$ and the second-order cone of dimension k by $\mathcal{Q}_k$, and is defined as

$$\mathcal{Q}_k = \{\mathbf{x} \in I\!R^k : \|\mathbf{x}_{2:k}\|_2 \leq x_1\},$$

(also called Lorentz cone, ice cream cone or quadratic cone), where $\|.\|_2$ stands for the Euclidean distance norm defined as $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$, $\forall \mathbf{x} \in I\!R^n$. The set of all $n \times n$ real Hankel matrices will be denoted by $\mathcal{H}_n$. An $n \times n$ real Hankel matrix H has the following structure:

$$H = \begin{bmatrix} h_1 & h_2 & \cdots & h_n \\ h_2 & h_3 & \cdots & h_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_n & h_{n+1} & \cdots & h_{2n-1} \end{bmatrix}.$$

It is clear that $\mathcal{H}_n \subset \mathcal{S}_n$. The Frobenius norm is defined on $\mathcal{S}_n$ as follows:

$$\|U\|_F = \sqrt{U \bullet U} = \|\mathbf{vec}^T(U)\mathbf{vec}(U)\|_2, \ \forall\, U \in \mathcal{S}_n \tag{1.1}$$

Here $U \bullet U = \text{trace}(U \cdot U) = \sum_{i,j}^n U_{i,j}^2$ and $\mathbf{vec}(U)$ stands for the vectorization operator found by stacking the columns of U together. The symbols $\succeq$ and $\geq_Q$ will be used to denote the partial orders induced by $\mathcal{S}_n^+$ and $\mathcal{Q}_k$ on $\mathcal{S}_n$ and $I\!R^k$, respectively. That is,

$$U \succeq V \quad \Leftrightarrow \quad U - V \in \mathcal{S}_n^+, \ \ \forall\, U,\ V \in \mathcal{S}_n$$

and

$$\mathbf{u} \geq_Q \mathbf{v} \Leftrightarrow \ \ \mathbf{u} - \mathbf{v} \in \ \mathcal{Q}_k, \ \ \forall\, \mathbf{u},\ \mathbf{v}\ \in I\!R^k.$$

The statement $\mathbf{x} \geq 0$ for a vector $\mathbf{x} \in I\!R^n$ means that each component of $\mathbf{x}$ is nonnegative. We use I and $\mathbf{0}$ for the identity and zero matrices. The dimensions of these matrices can be discerned from the context.

Our problem in mathematical notation can, now, be formulated as follows: Given a data matrix $F \in \mathbb{R}^{n\times n}$, find the nearest positive semidefinite Hankel matrix H to F such that $\|F - H\|_F$ is minimal. Thus, we have the following optimization problem:

$$\begin{aligned} \text{minimize} \ & \|F - H\|_F \\ \text{subject to} \ & H \in \mathcal{H}_n, \\ & H \succeq 0. \end{aligned} \tag{1.2}$$

It is worth describing the alternating projection method briefly; since this method is the most accurate, and converges to the optimal solution globally. However, the rate of convergence is slow. That makes it a good tool to provide us with accurate solutions against which we can compare the results obtained by the interior point methods. For these reason we devote Section 2 to the projection method. A brief description of semidefinite and second-order cone optimization problems along with reformulations of problem (1.2) in the form of the respective class will be given in Sections 3 and 4, respectively.

2 The projection Method

The method of successive cyclic projections onto closed subspaces C_i's was first proposed by von Neumann [19] and independently by Wiener [27]. They showed that if, for example, C_1 and C_2 are subspaces and D is a given point, then the nearest point to D in $C_1 \cap C_2$ could be obtained by the following algorithm:

Alternating Projection Algorithm
Let $X_1 = D$
for $k = 1, 2, 3, \ldots$
  $X_{k+1} = P_1(P_2(X_k))$.

X_k converges to the near point to D in $C_1 \cap C_2$, where P_1 and P_2 are the orthogonal projections on C_1 and C_2, respectively. Dykstra [10] and Boyle and Dykstra [7] modified von Neumann's algorithm to handle the situation when C_1 and C_2 are replaced by convex sets. Other proofs and connections to duality along with applications were given in Han [14]. These modifications were applied in [12] to find the nearest Euclidean distance matrix to a given data matrix. The modified Neumann's algorithm when applied to (1.2) yields the following algorithm, called the Modified Alternating Projection Algorithm: Given a data matrix F, we have:

Let $F_1 = F$
for $j = 1, 2, 3, \ldots$
  $F_{j+1} = F_j + [P_S(P_H(F_j)) - P_H(F_j)]$

Then $\{P_H(F_j)\}$ and $P_S(P_H(F_j))$ converge in Frobenius norm to the solution. Here, $P_H(F)$ is the orthogonal projection onto the subspace of Hankel matrices $\mathcal{H}_n$. It is simply setting each antidiagonal to be the average of the corresponding antidiagonal of F. $P_S(F)$ is the projection of F onto the convex cone of positive semidefinite symmetric matrices. One finds $P_S(F)$ by finding a spectral decomposition of F and setting the negative eigenvalues to zero.

3 Semidefinite Programming Approach

The semidefinite programming (SDP) problem in *primal standard form* is:

$$\begin{aligned} (P)\quad & \min_X \; C \bullet X \\ & \text{s. t.} \;\; A_i \bullet X = b_i, \quad i = 1, \cdots, m \\ & \qquad\;\; X \succeq 0, \end{aligned} \tag{3.1}$$

where all $A_i, C \in \mathcal{S}_n, b \in I\!R^m$ are given, and $X \in \mathcal{S}_n$ is the variable. This optimization problem is a convex optimization problem since its objective and constraints are convex.
We also consider SDP in *dual standard form*:

$$\begin{aligned} (D)\quad & \max_{\mathbf{y}} \; \mathbf{b}^T\mathbf{y} \\ & \text{s. t.} \\ & \qquad \sum_{i=1}^{m} y_i A_i + S = C \\ & \qquad S \succeq 0, \end{aligned} \tag{3.2}$$

where $\mathbf{y} \in I\!R^m$ and $S \in \mathcal{S}_n$ are the variables. This can be written as

$$\begin{aligned} & \max_{\mathbf{y}} \; \mathbf{b}^T\mathbf{y} \\ & \text{s. t.} \\ & \qquad \sum_{i=1}^{m} y_i A_i \preceq C, \end{aligned} \tag{3.3}$$

The second dual form (3.3) will be used throughout this section due to its simplicity and also the flexibility it provides for modeling.

Although the SDP problem (3.3) may appear quite specialized, it includes many important optimization problems as special cases. For instance, consider the linear program (LP)

$$\begin{aligned} & \min \;\; \mathbf{c}^T\mathbf{x} \\ & \text{s.t.} \;\; A\mathbf{x} + \mathbf{b} \geq 0 \end{aligned} \tag{3.4}$$

in which the inequality denotes component wise inequality. Since a vector $\mathbf{v} \geq 0$ if and only if $\mathrm{diag}(\mathbf{v}) \succeq 0$ (i.e., the diagonal matrix with the components of $\mathbf{v}$ on its diagonal) is positive semidefinite, we can express the LP (3.4) as a dual SDP problem (3.3) with

$$\mathbf{b} = \mathbf{c},\ C = -\mathrm{diag}(\mathbf{b}),\ A_i = \mathrm{diag}(a_i),\quad i = 1, \cdots, m;$$

where $A = [a_1, \cdots, a_m] \in I\!R^{n \times m}$.
To introduce other examples, we have to present the following useful theorem.

Theorem 3.1 (Schur Complement)
If

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

where $A \in \mathcal{S}_n^+$ and $C \in \mathcal{S}_n$, then the matrix M is positive (semi)definite if and only if the matrix $C - B^T A^{-1} B$ is positive (semi)definite. □

The matrix $C - B^T A^{-1} B$ is called the Schur complement of A in M.

Proof:
The result follows by setting $D = -A^{-1}B$, and noting that

$$\begin{bmatrix} I & 0 \\ D^T & I \end{bmatrix} \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \begin{bmatrix} I & D \\ 0 & I \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & C - B^T A^{-1} B \end{bmatrix}.$$

Since a block diagonal matrix is positive (semi)definite if and only if its blocks are positive (semi)definite, the proof is complete.
Now, we introduce the so-called general convex quadratically constrained quadratic program (QCQP)

$$\begin{array}{ll} \min & f_0(\mathbf{x}) \\ \text{s.t.} & f_i(\mathbf{x}) \leq 0, \quad i = 1, \cdots, L, \end{array} \tag{3.5}$$

where each f_i is a convex quadratic function $f_i(\mathbf{x}) = \mathbf{x}^T B_i \mathbf{x} - 2\mathbf{c}_i^T \mathbf{x} - d_i,\ B_i \in \mathcal{S}_n^+$. Assume for simplicity that $B_i \in \mathcal{S}_n^{++}$, hence, let $B_i = B_i^{1/2} B_i^{1/2}$. Then using Theorem 3.1, Problem (3.5) can be written as

$$\begin{array}{ll} \min & t \\ \text{s.t.} & \begin{bmatrix} I & B_i^{1/2}\mathbf{x} \\ (B_i^{1/2}\mathbf{x})^T & \mathbf{c}_i^T \mathbf{x} + d_i + t \end{bmatrix} \succeq 0, \\ & \begin{bmatrix} I & B_i^{1/2}\mathbf{x} \\ (B_i^{1/2}\mathbf{x})^T & \mathbf{c}_i^T \mathbf{x} + d_i \end{bmatrix} \succeq 0. \end{array}$$

which is an SDP problem in the dual form with $\mathbf{x} \in I\!R^n$ and $t \in I\!R$ as variables. This SDP problem has dimensions $n+1$ and $L \times n + L + 1$.

Inspired by the above examples, we will formulate Problem (1.2) as an SDP problem in the dual form (3.3). To do so, we need to use some tools. The following theorem, which can be considered as a corollary of Theorem (3.1), provide these tools.

Theorem 3.2
Let $\mathbf{a}(x) \in I\!R^n$ depend affinely on x. Then the following minimization problem:

$$min \quad \|\mathbf{a}(x)\|_2,$$

cab be solved by solving the following SDP problem:

$$min \ \ t, \ \ s.t. \ \begin{bmatrix} I & \mathbf{a}(x) \\ (\mathbf{a}(x))^T & tI \end{bmatrix} \succeq 0,$$

where t is a nonnegative real scalar. □

Proof:
Since $\|\mathbf{a}\|_2 = \sqrt{\mathbf{a}^T\mathbf{a}}$, we may minimize $\|\mathbf{a}\|_2$ by minimizing $\mathbf{a}^T\mathbf{a}$. So, let $\mathbf{a}^T\mathbf{a} \leq t$. Hence, $tI - \mathbf{a}^T I \mathbf{a} \succeq 0$. So, by Theorem (3.1) the proof is complete.

3.1 SDV Formulation

We are now ready to introduce the first formulation of (1.2) as an SDP problem. We have

$$\|F - H\|_F = \|\mathbf{vec}(F - H)\|_2$$

So by Theorem 3.2, Problem (1.2) is cast as

$$\begin{aligned} (SDV) \quad & \min \ t \\ & \text{s.t.} \\ & \quad \begin{bmatrix} t & 0 & 0 \\ 0 & H & 0 \\ 0 & 0 & V \end{bmatrix} \succeq 0, \end{aligned} \tag{3.6}$$

where

$$V = \begin{bmatrix} I & \mathbf{vec}(F - H) \\ \mathbf{vec}^T(F - H) & t \end{bmatrix}$$

and $t \in I\!R^+$. Problem (3.6) is an SDP problem in dual form (3.3) with dimensions $2n$ and n^2+n+2. To see this, we identify

$$
\begin{aligned}
& y_1 = t, \;\; y_k = h_{k-1}, \quad k = 2, \cdots, 2n \\
& \mathbf{b} = \begin{bmatrix} -1 & 0 & \cdots & 0 \end{bmatrix}^T, \\
& A_1 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \\
& A_k = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -E_{k-1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{vec}(E_{k-1}) \\ 0 & 0 & \mathbf{vec}^T(E_{k-1}) & 0 \end{bmatrix}, \; k = 2, \cdots, 2n \\
& C = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I & \mathbf{vec}(F) \\ 0 & 0 & \mathbf{vec}^T(F) & 0 \end{bmatrix},
\end{aligned}
$$

The matrices E_{k-1} are defind as follows

$$
E_k(i,j) = \begin{cases} 1 & \text{if } \; i+j = k+1, \\ 0 & \text{otherwise.} \end{cases} \tag{3.7}
$$

which form an orthonormal basis for $\mathcal{H}_n$, where where (i,j)-entry of each E_k

To illustrate how can we use the formulation SDV to model Problem (1.2), we consider the following example:

Example 3.1
Consider Problem (1.2) with

$$
F = \begin{bmatrix} -4 & 2 & 1 \\ -6 & -1 & 0 \\ 3 & 2 & 7 \end{bmatrix}
$$

and let $t \in I\!R^+$. Then we have

- *SDV: The SDV formulation is*

$$
\begin{aligned}
&\min\ t\\
&s.t.\\
&\begin{bmatrix}
t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & h_2 & h_3 & h_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & h_3 & h_4 & h_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_1\\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_2\\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & s_3\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & s_4\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & s_5\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & s_6\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & s_7\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & s_8\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & s_9\\
0 & 0 & 0 & 0 & s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 & s_9 & t
\end{bmatrix} \succeq 0,
\end{aligned}
$$

where

$$
\boldsymbol{s} = [s_i]_{i=1}^{9} = \begin{bmatrix} -4-h_1\\ -6-h_2\\ 3-h_3\\ 2-h_2\\ -1-h_3\\ 2-h_4\\ 1-h_3\\ -h_4\\ 7-h_5 \end{bmatrix}
$$

or equivalently,

$$
\begin{aligned}
&\max\ \ -t\\
&s.t.\\
&A_1 t + A_2 h_1 + A_3 h_2 + A_4 h_3 + A_5 h_4 + A_6 h_5 \preceq C
\end{aligned}
$$

where

$$A_1 = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

and for $k = 2, \cdots, 6$ *we have*

$$A_k = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -E_{k-1} & 0 & 0 \\ 0 & 0 & I & \boldsymbol{vec}(E_{k-1}) \\ 0 & 0 & \boldsymbol{vec}^T(E_{k-1}) & 0 \end{bmatrix}$$

Finally

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 0 & -4 & -6 & 3 & 2 & -1 & 2 & 1 & 0 & 7 & 0 \end{bmatrix}$$

4 Mixed Semidefinite and Second-Order Cone Approach:

We consider the *second-order cone program* (SOCP)

$$\begin{aligned} &\min \ f^T\mathbf{x} \\ &\text{s.t.} \ \|A_i\mathbf{x}+\mathbf{b}_i\| \le \mathbf{c}_i^T\mathbf{x}+d_i, \quad i=1,\cdots,N, \end{aligned} \tag{4.1}$$

where $\mathbf{x} \in I\!R^n$ is the optimization variable, and $f \in I\!R^n$, $A_i \in I\!R^{(n_i-1)\times n}$, $\mathbf{b}_i \in I\!R^{n_i-1}$, $\mathbf{c}_i \in I\!R^n$, and $d_i \in I\!R$ are given data. The norm appearing in the constraint is the standard Euclidean norm, *i. e.* , $\|\mathbf{u}\| = \left(\mathbf{u}^T\mathbf{u}\right)^{1/2}$. We call the constraint

$$\|A_i\mathbf{x}+\mathbf{b}_i\| \le \mathbf{c}_i^T\mathbf{x}+d_i,$$

a *second-order cone constraint of dimension* n_i, simply because

$$\|A_i\mathbf{x}+\mathbf{b}_i\| \le \mathbf{c}_i^T\mathbf{x}+d_i \iff \begin{bmatrix}\mathbf{c}_i^T\\ A_i\end{bmatrix}\mathbf{x}+\begin{bmatrix}d_i\\ \mathbf{b}_i\end{bmatrix} \in \mathcal{Q}_{n_i}$$

Recall that a second-order cone of dimension n_i is defined as

$$\mathcal{Q}_{n_i} = \left\{ \begin{bmatrix} t\\ \mathbf{u}\end{bmatrix} : \mathbf{u} \in I\!R^{n_i-1},\ t \in I\!R,\ \|\mathbf{u}\| \le t \right\},$$

and hence the set of points satisfying a second-order cone constraint is convex. Thus, the SOCP (4.1) is a convex programming problem since the objective is convex function and the constraints define a convex set.
Second-order cone constraints can be used to represent several common convex constraints. For example, when $n_i = 1$ for $i = 1,\cdots,N$, the SOCP (4.1) reduces to the LP problem:

$$\begin{aligned} &\min \ f^T\mathbf{x} \\ &\text{s.t.} \ 0 \le \mathbf{c}_i^T\mathbf{x}+d_i, \quad i=1,\cdots,N. \end{aligned}$$

Another interesting example is convex quadratically constrained quadratic program (QCQP) (3.5, page 6). In this example we have the problem

$$\begin{aligned} &\min \ \mathbf{x}^TB_0\mathbf{x}-2\mathbf{c}_0^T\mathbf{x}-d_0 \\ &\text{s.t.} \ \mathbf{x}^TB_i\mathbf{x}-2\mathbf{c}_i^T\mathbf{x}-d_i \le 0, \quad i=1,\cdots,L. \end{aligned} \tag{4.2}$$

This problem can be rewritten as

$$\begin{aligned} &\min \ \|B_0^{1/2}\mathbf{x}-B_0^{-1/2}\mathbf{c}_0\|^2-d_0+\mathbf{c}_0^TB_0^{-1}\mathbf{c}_0 \\ &\text{s.t.} \ \|B_i^{1/2}\mathbf{x}-B_i^{-1/2}\mathbf{c}_i\|^2-d_i+\mathbf{c}_i^TB_i^{-1}\mathbf{c}_i \le 0, \quad i=1,\cdots,L, \end{aligned}$$

which can be solved via the SOCP with $L+1$ constraints of dimension $n+1$

$$\begin{aligned} &\min\ t \\ &\text{s.t.}\ \ \|B_0^{1/2}\mathbf{x} - B_0^{-1/2}\mathbf{c}_0\| \le t, \\ &\qquad \|B_i^{1/2}\mathbf{x} - B_i^{-1/2}\mathbf{c}_i\| \le (d_i - \mathbf{c}_i^T B_i^{-1}\mathbf{c}_i)^{1/2},\ \ i = 1,\cdots,L, \end{aligned} \tag{4.3}$$

where $t \in I\!R$ is a new optimization variable. Problems (3.5) and (4.3) will have the same optimal solution and the same optimal values up to a constant.

This shows that SOCP contains interesting examples. On the other hand, it is itself contained in SDP. This can be seen by the following property which is true for each vector $\mathbf{u}$ and scalar t:

$$\|\mathbf{u}\| \le t \iff \begin{bmatrix} tI & \mathbf{u} \\ \mathbf{u}^T & t \end{bmatrix} \succeq 0,$$

using this property SOCP (4.1) can be expressed as SDP

$$\begin{aligned} &\min\ f^T\mathbf{x} \\ &\text{s.t.}\ \begin{bmatrix} (\mathbf{c}_i^T\mathbf{x} + d_i)I & A_i\mathbf{x} + b_i \\ (A_i\mathbf{x} + b_i)^T & \mathbf{c}_i^T\mathbf{x} + d_i \end{bmatrix} \succeq 0,\ \ i = 1,\cdots,N. \end{aligned} \tag{4.4}$$

Solving SOCP via SDP is not a good idea, however. Interior point methods that solve the SOCP directly have a much better worst-case complexity than an SDP method applied to (4.1): the number of iterations to decrease the duality gap to a constant fraction of itself is bounded above by $\mathcal{O}(\sqrt{N})$ for the SOCP algorithm, and by $\mathcal{O}(\sqrt{\sum_i n_i})$ for the SDP algorithm (see [17]). More importantly in practice, each iteration is much faster: the amount of work per iteration is $\mathcal{O}(n^2 \sum_i n_i)$ in the SOCP algorithm and $\mathcal{O}(n^2 \sum_i n_i^2)$ for the SDP. The difference between these numbers is significant if the dimensions n_i of the second-order constraints are large.

Returning to Problem (4.1) we see that it may be written in such a way to have a constraint over the positive semidefinite cone and a constraint over the second-order cone, namely: $H \succeq 0$ and $\|F - H\|_F \le t$, respectively. Thus, in order to take advantage of the good behavior of the SOCP we need to deal with mixed SDP and SOCP problems. Fortunately, interior point methods that solve such mixed problems are available. Indeed, these methods are the same as those of SDP with slight modifications. We will discuss these methods in the coming chapter. But now let us study our problem in a more general framework.

A *cone-linear programming* (Cone-LP) is a unified way to study SDP and SOCP problems. The standard canonical form of Cone-LP problems is

$$\min\ \mathbf{c}^T\mathbf{x}\ \text{ s.t. }\ A\mathbf{x} = \mathbf{b},\ \mathbf{x} \in \mathcal{K}, \tag{4.5}$$

where $\mathbf{x} \in I\!R^n$ is the vector of decision variables, $\mathcal{K} \subset I\!R^n$ is a pre-specified convex cone, and $\mathbf{b} \in I\!R^m$, $\mathbf{c} \in I\!R^n$ and $A \in I\!R^{m \times n}$ are given data. Despite its name, Cone-LP is non-linear, since $\mathcal{K}$ need not be polyhedral.

Important subclasses of Cone-LP are linear programming, semidefinite programming, second-order cone programming, and a mixture of these. These subclasses arise by letting $\mathcal{K}$ in (4.5) be the nonnegative orthant $\mathcal{K} = I\!R^n_+$, the cone of positive semidefinite matrices $\mathcal{S}_n^+$, the second-order cone $\mathcal{Q}_n$, or a mixture of them, respectively.

A mixed semidefinite and second-order cone optimization problem can be formulated as a standard Cone-LP problem (4.5) with the following structure:

$$\begin{aligned} \min\ & C_S \bullet X_S + C_Q^T X_Q + C_L^T X_L \\ \text{s.t.}\ & (A_S)_i \bullet X_S + (A_Q)_i^T X_Q + (A_L)_i^T X_L = b_i, \quad i = 1, \cdots, m \\ & X_S \succeq 0, X_S \geq_Q 0, X_L \geq 0, \end{aligned} \tag{4.6}$$

where $X_S \in \mathcal{S}_n$, $X_Q \in I\!R^k$ and $X_L \in I\!R^{n_L}$ are the variables. C_S, $(A_S)_i \in \mathcal{S}_n$, $\forall i$, C_Q, $(A_Q)_i \in I\!R^k$ $\forall i$ and C_L, $(A_L)_i \in I\!R^{n_L}$ $\forall i$ are given data. Each of the three inequalities has a different meaning: $X_S \succeq 0$ means, as we have seen, that $X_S \in \mathcal{S}_n^+$, $X_S \geq_Q 0$ means that $X_Q \in \mathcal{Q}_k$ and $X_L \geq 0$ means that each component of X_L is nonnegative. It is possible that one or more of the three parts of (4.6) is not present. If the second-order part is not present, then (4.6) reduces to the ordinary SDP (3.1) and if the semidefinite part is not present, then (4.6) reduces to the so-called convex quadratically constrained linear programming problem. The standard dual of (4.6) is:

$$\begin{aligned} \max\ & \mathbf{b}^T \mathbf{y} \\ \text{s.t.}\ & \\ & \sum_{i=1}^m y_i (A_S)_i \preceq C_S \\ & \sum_{i=1}^m y_i (A_Q)_i \leq_Q C_Q \\ & \sum_{i=1}^m y_i (A_L)_i \leq C_L. \end{aligned} \tag{4.7}$$

Here, $\mathbf{y} \in I\!R^m$ is the variable. The dual problem is interesting because it provides a machinery to formulate many problems in a natural manner.

In this context, we may drop the third part of the constraints in (4.6) and its dual (4.7), since we do not have explicit linear constraints. In the remaining of this chapter we discuss different ways to put Problem (1.2) in the form of (4.7).

As a matter of fact, we can do that in three different ways depending on how we define the Frobenius norm $\|F - H\|_F$.

4.1 SQV Formulation

One way to define $\|F - H\|_F$ is

$$\|F - H\|_F = \|\mathbf{vec}(F - H)\|_2.$$

So, if we put $\|F - H\|_F \leq t$ for $t \in I\!R^+$, then by the definition of the second-order cone, we have

$$\begin{bmatrix} t \\ \mathbf{vec}(F - H) \end{bmatrix} \in \mathcal{Q}_{1+n^2}.$$

Hence, we have the following reformulation of (1.2):

$$\begin{aligned} (SQV) \quad & \min \quad t \\ & \text{s.t.} \quad \begin{bmatrix} t & 0 \\ 0 & H \end{bmatrix} \succeq 0, \\ & \qquad \begin{bmatrix} t \\ \mathbf{vec}(F - H) \end{bmatrix} \geq_Q 0. \end{aligned} \tag{4.8}$$

This problem is in the form of (4.7) with

$$\mathbf{b} = [-1 \; 0 \; \cdots \; 0]^T,$$

$$(A_S)_1 = \begin{bmatrix} -1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad (A_S)_k = \begin{bmatrix} 0 & 0 \\ 0 & -E_{k-1} \end{bmatrix}, \quad k = 2, \cdots, 2n-1,$$

$$(A_Q)_1 = \begin{bmatrix} -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad (A_Q)_k = \begin{bmatrix} 0 \\ \mathbf{vec}(E_{k-1}) \end{bmatrix}, \quad k = 2, \cdots, 2n-1,$$

$$C_S = 0_{(n+1)\times(n+1)}, \qquad C_Q = \begin{bmatrix} 0 \\ \mathbf{vec}(F) \end{bmatrix},$$

where the matrices E_{k-1}'s are defined in (3.7), page 8. Although this formulation is natural and straightforward, we notice that the dimension of the second-order cone constraint is large, $1 + n^2$. Also, in this formulation the special structure of Problem (1.2) is not fully exploited. Hence we should look for another way of formulation which exploits the structure and has a dimension of less magnitude. The SDP part of any mixed formulation will be the same as above. However, the second-order cone part will make the difference; since it is what we can manipulate.

References

[1] S. Al-Homidan. Hybrid methods for approximating Hankel matrix. *Numerical Algorithms*. To appear.

[2] S. Al-Homidan. Combined methods for apprpximating Hankel matrix. *WSEAS Transactions on systems*, 1:35–41, 2002.

[3] F. Alizadeh and D. Goldfarb. Second-order cone programming. Technical Report RRR Report 51-2001, RUTCOR, Rutgers University. 6, 2001.

[4] F. Alizadeh, J. A. Haeberly, M. V. Nayakkanakuppann, M. Overton, and S. Schmieta. SDPPack, user's guide, 1997.

[5] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton. Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results. *SIAM J. Optim.*, 8:746–768, 1998.

[6] R. Bellman and K. Fan. On systems of linear inequalities in Hermitian matrix variables. In V. L. Klee, editor, *Convexity*, volume 7, pages 1–11. Proc. Symposia in Pure Mathematics, Amer. Math. Soc., Providence,RI, 1963.

[7] J. P. Boyle and R. L. Dykstra. A method of finding projections onto the intersection of convex sets in Hilbert space. *Lecture Notes in Statistics*, 37:28–47, 1986.

[8] J.P. Burg, D. G. Luenberger, and D. L. Wenger. Estimation of structured covariance matrices. *Proc. IEEE*, 70:963–974, 1982.

[9] E. de Klerk. *Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications*, volume 65 of *Applied Optimization Series*. Kluwer Academic Publishers, 2002.

[10] R. L. Dykstra. An algorithm for restricted least squares regression. *J. Amer. Stat.*, 78:839–842, 1983.

[11] W. Fang and A.E. Yagle. Two methods of Toeplitz-plus-Hankel approximation to a data covariance matrix. *IEEE Trans. Signal Processing*, 40:1490–1498, 1992.

[12] W. Glunt, L. Hayden, S. Hong, and L. Wells. An alternating projection algorithm for computing the nearest Euclidean distance matrix. *SIAM J. Matrix Anal. Appl.*, 11(4):589–600, 1990.

[13] K. M. Grigoriadis, A. E. Frazho, and R. E. Skelton. Application of alternating convex projection methods for computing of positive Toeplitz matrices. *IEEE Trans. Signal Processing*, 42:1873–1875, 1994.

[14] S. P. Han. A successive projection method. *Math. Programming*, 40:1–14, 1988.

[15] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and Applications*, (284):193–228, 1998.

[16] C. S. Macinnes. The solution to a structured matrix approximation problem using Grassman coordinates. *SIAM J. Matrix Anal. Appl.*, 211(2):446–453, 1999.

[17] Y. Nesterov and A. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming.* SIAM, Philadelphia, 1994.

[18] Yu. E. Nestrov and M. J. Todd. Primal-dual interior-point methods for self-scaled cones. *SIAM J. Optim.*, 8:324–364, 1998.

[19] J. Von Neumman. *Functional Operators II, The geometry of orthogonal spaces.* Annals of Math. studies No.22, Princeton Univ. Press., 1950.

[20] J. Sturm. Implementation of interior point methods for mixed semidefinite and second order cone optimization problems. Technical report, August 2002.

[21] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

[22] Y. J. Suffridge and T. L. Hayden. Approximation by a Hermitian positive semidefinite Toeplitz matrix. *SIAM J. Matrix Analysis and Appl.*, 14:721–734, 1993.

[23] R. Tütüncü, K. Toh, and M. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. to appear.

[24] M. J. Todd. Semidefinite Optimization. *Acta Numerica*, 10:515–560, 2001.

[25] M. J. Todd, K. C. Toh, and R. H. Tütüncü. SDPT3 — a Matlab software package for semidefinite programming. *Optim. Methods Softw.*, 11:545–581, 1999.

[26] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Rev.*, 38(1):49–95, 1996.

[27] N. Wiener. On factorization of matrices. *Comm. Math. Helv.*, 29:97–111, 1955.

[28] H. Wolkowicz, R. Saigal, and L. Vandenberghe. *Handbook of Semidefinite Programming.* Kluwer Academic Publishers Group, Boston-Dordrecht-London, 2000.