SDP formulation

We write our standard form semidefinite program as

$$\begin{align*}
\min_X & \quad C \cdot X \\
\text{subject to} & \quad A_i \cdot X = b_i, \ i = 1, \ldots, m \\
& \quad X \succeq 0
\end{align*}$$

where:
- $C$ is an $n \times n$ symmetric matrix
- $A_i$ is an $n \times n$ symmetric matrix for $i = 1, \ldots, m$
- $b_i$ is a scalar for $i = 1, \ldots, m$.

The parameter matrices $C$ and $A_i$ need not be positive semidefinite, although they are assumed to be symmetric. Recall that $C \cdot X$ represents the Frobenius inner product between the symmetric matrices $C$ and $X$, which is equal to the trace($CX$). The dual problem is

$$\begin{align*}
\max_{y \in \mathbb{R}^m, S \in \mathbb{S}_+^n} & \quad b^T y \\
\text{subject to} & \quad \sum_{i=1}^m y_i A_i + S = C \\
& \quad S \succeq 0
\end{align*}$$

Notice that the dual slack variables $S = C - \sum_{i=1}^m y_i A_i$ constitute a symmetric positive semidefinite matrix in feasible dual solutions.

Barrier function

The following is a standard theorem on matrices.

**Theorem 1.** Let $A$ be a symmetric $n \times n$ matrix. The trace of $A$ is equal to the sum of the eigenvalues of $A$. The determinant of $A$ is equal to the product of the eigenvalues of $A$. The matrix is positive semidefinite if and only if all its eigenvalues are nonnegative. It is positive definite if and only if all the eigenvalues are positive.

We used the barrier function $-\sum_{i=1}^n \ln x_i$ for linear programming, with the barrier increasing as $x$ approaches the boundary of the nonnegative orthant.

For semidefinite programming, the eigenvalues are positive on the interior of the cone $\mathbb{S}_+^n$, and one or more of them approaches zero as $X$ approaches the boundary of the cone.

This analogy suggests the use of a barrier function consisting of $-\sum_{i=1}^n \ln \lambda_i$, where the $\lambda_i$ are the eigenvalues of the matrix $X$. By exploiting Theorem 1, we can write the barrier function as

$$\phi(X) = -\ln(\det(X)) = -\ln(\Pi_{i=1}^n \lambda_i) = \sum_{i=1}^n \ln \lambda_i.$$  \hspace{0.5cm} (3)
Introducing the barrier function into (1) gives the barrier problem:

\[
\min_X \quad C \cdot X - \mu \ln(\det(X)) \\
\text{subject to } \quad A_i \cdot X = b_i, \; i = 1, \ldots, m \\
X \succeq 0
\]  

(4)

We let \( f(X) \) denote the objective function so

\[
f(X) := C \cdot X - \mu \ln(\det(X)).
\]  

(5)

The gradient of this function is

\[
\nabla f(X) = C - \mu X^{-1}
\]  

(6)

Note that we’ve represented the gradient as a \( n \times n \) symmetric matrix, since \( X \) is an \( n \times n \) symmetric matrix. This leads to the Karush-Kuhn-Tucker conditions for (4):

\[
C - \mu X^{-1} - \sum_{i=1}^{m} y_i A_i = 0
\]  

(7)

\[
A_i \cdot X = b_i, \; i = 1, \ldots, m
\]  

(8)

Equivalently, we can write this as

\[
\sum_{i=1}^{m} y_i A_i + S = C
\]  

(9)

\[
A_i \cdot X = b_i, \; i = 1, \ldots, m
\]  

(10)

\[
S - \mu X^{-1} = 0
\]  

(11)

Interior point path-following methods can be constructed using a Newton system to find a direction at each iteration. The barrier parameter \( \mu \) can be updated in a similar manner to path following methods for LP. The barrier function possesses a property known as self-concordance, which requires the third derivative to be bounded as a function of the second derivative. If a barrier function is self-concordant then a path-following method can be constructed that converges in a polynomial number of iterations. A short step path following method requires \( O(\sqrt{n}) \) iterations. (This construction and convergence result is due to Nesterov and Nemirovski.)
Setting up a Newton system to find directions

The last condition (11) could be written equivalently as

\[ XS = \mu I. \]  
(12)

Setting up the Newton system for this equation directly would result in the condition

\[ X\Delta S + S\Delta X = \mu I - XS \]  
(13)

However, since \( X \) and \( S \) are not diagonal matrices, this system will result in directions \( \Delta X \) and \( \Delta S \) that are not symmetric. Thus, (12) is represented in different and equivalent ways to ensure the directions are symmetric. The general framework is

\[
\frac{1}{2} \left( PXS^{-1} + P^{-T}SXP^T \right) = \mu I.
\]  
(14)

Many different choices have been proposed for \( P \). For example:

- \( P = I \): Alizadeh, Haeberly, Overton
- \( P = X^{-1/2} \): independently by three groups of researchers, known as the H..K..M update, is the choice in most interior point SDP solvers.
- \( P = S^{1/2} \)
- The general formulation is known as the Nesterov-Todd formulation.

The Newton system then becomes

\[
\sum_{i=1}^{m} \Delta y_i A_i + \Delta S = -r_C
\]  
(15)

\[
A_i \cdot \Delta X - b_i = -r_b, \ i = 1, \ldots, m
\]  
(16)

\[
\frac{1}{2} \left( P\Delta XSP^{-1} + P^{-T}S\Delta XP^T \right) + \frac{1}{2} \left( PXS^{-1} + P^{-T}SXP^T \right) = \mu I - \frac{1}{2} \left( PXS^{-1} + P^{-T}SXP^T \right)
\]  
(17)

where \( r_C \) and \( r_b \) denote the dual and primal residuals respectively. This is clearly a more complicated system than that for LP, but the algorithmic structure is similar.
Packages

Primal-dual interior point packages for SDP include:

- SDPT3
- SeDuMi
- CSDP

These methods can solve problems with $n$ in the hundreds.

Alternative approaches

The linear algebra overhead at each iteration of an interior point method for semidefinite programming can be prohibitive for larger problems. Various alternative approaches have been proposed.

- SDPLR: Working with a low rank factorization. We write $X = LL^T$, where the entries of $L$ are the variables. The matrix $L$ has only a limited number of columns. The number of variables is smaller, but there are quadratic equality constraint from $A_i \bullet LL^T = b_i$. The package uses an augmented Lagrangian approach.

- Spectral bundle approaches: Work with a dynamically modified low-rank approximation to $X$. Use $X = PV^T + \alpha W$, where the variables at each iteration are $V$ and $\alpha$. The matrices $P$ and $W$ are dynamically updated depending on the update to $X$. The dimension of $V$ is considerably smaller than that of $X$.

- Cutting plane and cutting surface methods have been proposed. These methods work with relaxations of the dual problem, imposing linear and second order cone constraints on $y$, rather than the original SDP constraint.

- Alternating methods working with projections: can choose a direction and an update, and then project onto $S^n_+$. The projection requires truncation: we only keep nonnegative eigenvalues and the corresponding eigenvectors.