Semidefinite Programming
Lecture 5

scribe: Ali Shokoufandeh

Feb. 27, 1995

1 Minimization of weighted sum of eigen values

Consider the following problem

$$\begin{array}{ll} \min & m_1\lambda_1 + \cdots + m_k\lambda_k(A(\mathbf{x})) \\ \text{s.t.} & A(\mathbf{x}) = A_0 + x_1A_1 + \cdots + x_mA_m \\ & m_1 \geq \cdots \geq m_k. \end{array}$$

¿From $m_1\lambda_1 + \cdots + m_k\lambda_k = (m_1 - m_2)\lambda_1 + (m_2 - m_3)(\lambda_2 + \lambda_3) + \cdots + (m_{k-1} - m_k)(\lambda_1 + \cdots + \lambda_{k-1}) + m_k)(\lambda_1 + \cdots + \lambda_k)$ and the fact that a positive combination of convex functions is convex, it follows that the objective function of this problem is a convex function.
This problem can be stated in the form of an SDP for a fixed matrix A as follows

$$\begin{array}{lll} \min & (m_1 - m_2)Y_1 \bullet A + (m_2 - m_3)Y_2 \bullet A + \cdots + m_kY_k \bullet A & \\ \text{s.t.} & \text{trace } (Y_i) = i & \text{for } i = 1, \cdots, k \\ & 0 \preceq Y_i \preceq I & \end{array}$$

with the dual

$$\begin{array}{ll} \min & \sum_{i=1}^k iZ_i + \sum_{i=1}^k \text{trace } (V_i) \\ \text{s.t.} & Z_iI + V_i - (m_i - m_{i+1})A \succeq 0 \quad \text{for } i = 1, \cdots, k \\ & V_i \succeq 0. \end{array}$$

Now replacing A by $A(\mathbf{x}) = A_0 + x_1A_1 + \cdots + x_mA_m$ yields the following SDP formulation of the problem

$$\begin{array}{ll} \min & \sum_{i=1}^k iZ_i + \sum_{i=1}^k \text{trace } (V_i) \\ \text{s.t.} & Z_iI + V_i - (m_i - m_{i+1})\sum_{j=1}^m x_jA_j \succeq (m_i - m_{i+1})A_0 \quad \text{for } i = 1, \cdots, k \\ & V_i \succeq 0 \end{array}$$

2 Interior method for SDP

2.1 Barrier functions. Given a convex set S in R^n, a function $b : \text{Int}(S) \rightarrow R$ is said to be a barrier function if it satisfies the following conditions:

1. b is convex over S.

2. Given any sequence of interior points $x_1, x_2, \ldots$ in S that converges to a point on the boundary of S, $\lim b(x_k) \to \infty$ as $k \to \infty$. More precisely the function $b(x)$ attains its minimum somewhere in the interior of S.

3. $b(x) \geq 0$ for all x in S.

Given a constrained minimization problem of a convex function $f(x)$ over S,we can always translate the problem into an unconstrained minimization problem of a new function f$(x)+\mu b(x)$ for $\mu \geq 0$ over S. Observe that the new objective function is still convex, and according to the definition of $b(x)$ the minimum of this new function always occurs on an interior point of S.
For a fixed value of μ define $x_\mu = \operatorname{argmin} f(x) + \mu b(x)$ over S. Denote by x^* a minimizer of $f(x)$ over S; intuitively it is clear that $x_\mu \to x^*$ as $\mu \to 0$.
Here are some examples of barrier functions:

1. Let S be the nonnegative orthant, then $b_1(\mathbf{x}) = \sum\limits_{i=1}^{n} \frac{1}{x_i}$ and $b_2(\mathbf{x}) = -\sum_{i=1}^{n} \ln x_i$ are two possible barriers for S.

2. Let S to be the set of PSD matrices then analogous to $b_1(x)$ for the positive orthant we can define

$$b_1(X) = \sum_{i=1}^{n} \frac{1}{\lambda_i(X)} = \frac{\sum\limits_{i=1}^{n} \prod\limits_{i \neq j} \lambda_j(X)}{\prod\limits_{i=1}^{n} \lambda_i}$$

   Observe that the denominator of the last relation refers to the determinant of X, and the numerator refers to the coefficient of characteristic polynomial one after the determinant. Also, analogous to the second barrier in previous example, one can define

$$b_2(X) = -\sum_i \ln \lambda_i(X) = \ln(\det(X))$$

   .

2.2 Newton's method for solving LP and SDP problems. Consider the linear programming problem in standard form

$$\begin{array}{lll} \mathbf{P}: & \min & \mathbf{c}^T\mathbf{x} \\ & \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & & \mathbf{x} \geq 0 \end{array}$$

Adding the barrier function we get the new problem

$$\begin{array}{lll} \mathbf{P}_\mu: & \min & \mathbf{c}^T\mathbf{x} - \mu \sum_{i=1}^{n} \ln \mathbf{x}_i \\ & \text{s.t.} & A\mathbf{x} = \mathbf{b}. \end{array}$$

Defining the Lagrangian as

$$L(\mathbf{x}, \mathbf{y}) = \mathbf{c}^T\mathbf{x} - \mu \sum_{i=1}^{n} \ln \mathbf{x}_i + \mathbf{y}^T(\mathbf{b} - A\mathbf{x})$$

the following optimality conditions follow

$$\begin{cases} \nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{y}) = \mathbf{c}^T - \mu(1/\mathbf{x}_1, \ldots, 1/\mathbf{x}_n)^T - \mathbf{y}^T A = 0 \\ \nabla_{\mathbf{y}} L(\mathbf{x}, \mathbf{y}) = \mathbf{b} - A\mathbf{x} = 0. \end{cases}$$

That yeilds to the following system

$$\begin{cases} A^T\mathbf{y} + \mathbf{z} = \mathbf{c} \\ A\mathbf{x} = \mathbf{b} \\ \mathbf{z}_i - \frac{\mu}{\mathbf{x}_i} = 0 \qquad \text{for } i = 1, \cdots, n. \end{cases} \tag{2.1}$$

For any value of $\mu \geq 0$, let $(\mathbf{x}_\mu, \mathbf{y}_\mu, \mathbf{z}_\mu)$ be a solution for system 2.1. The sequence of points $\{(\mathbf{x}_\mu, \mathbf{y}_\mu, \mathbf{z}_\mu)\}$ converges to an optimal solution of the linear programming problem as $\mu \to 0$, if one exists. The set $\{\mathbf{x}_\mu | \mathbf{x}_\mu$ is the solution of $\mathbf{P}_\mu$ for $\mu > 0\}$ is known as the primal central path.
Writing the dual problem as

$$\begin{array}{lll} \mathbf{D}: & \max & \mathbf{b}^T\mathbf{y} \\ & \text{s.t.} & A^T\mathbf{y} + \mathbf{z} = \mathbf{c} \\ & & \mathbf{z} \geq 0 \end{array}$$

and adding the barrier we get

$$\begin{array}{lll} \mathbf{D}_\nu: & \max & \mathbf{b}^T\mathbf{y} + \nu \sum_i \ln \mathbf{z}_i \\ & \text{s.t.} & A^T\mathbf{y} + \mathbf{z} = \mathbf{c}. \end{array}$$

The Lagrangian of this problem can be stated as

$$L(\mathbf{y}, \mathbf{z}, \mathbf{x}) = \mathbf{b}^T\mathbf{y} + \nu \sum_{i=1}^{n} \ln \mathbf{z}_i + \mathbf{x}^T(\mathbf{c} - A^T\mathbf{y} - \mathbf{z})$$

and we get the following optimality conditions for D_ν

$$\begin{cases} \nabla_{\mathbf{y}} L(\mathbf{y}, \mathbf{z}, \mathbf{x}) = & \mathbf{b} - A\mathbf{x} = 0 \\ \nabla_{\mathbf{z}} L(\mathbf{y}, \mathbf{z}, \mathbf{x}) = & \nu(1/\mathbf{z}_1, \ldots, 1/\mathbf{z}_n)^T - \mathbf{x}^T = 0 \\ \nabla_{\mathbf{y}} L(\mathbf{y}, \mathbf{z}, \mathbf{x}) = & \mathbf{c} - A^T\mathbf{y} - \mathbf{z} = 0. \end{cases} \tag{2.2}$$

As in the case of the primal problem, one can define the dual central path as

$$\{(\mathbf{y}_\nu, \mathbf{z}_n u) | (\mathbf{y}_\nu, \mathbf{z}_n u) \text{ is the solution of } \mathbf{D}_\nu \text{ for } \nu > 0\},$$

and equivalently the primal dual central path as

$$\{(\mathbf{y}_\mu, \mathbf{z}_\mu, \mathbf{z}_\mu) | A\mathbf{x}_\mu = \mathbf{b},\ A^T\mathbf{y}_m u + \mathbf{z}_m u = \mathbf{c},\ (\mathbf{x}_\mu)_i(\mathbf{z}_\mu)_i = \mu,\ \mu \geq 0\}.$$

Clearly the structure of central paths depends on the barrier functions used in P_μ or D_ν. For the case of linear programming, the central path is a smooth curve. A family of polynomial algorithms for linear programming, tries to approximate central path with a polygonal path that stays close to central path, and following the path in direction in which $\mu \to 0$. A major factor in the complexity of these algorithms is the number of breaks in the approximated polygonal path.

Next, we can use the general Newton method to solve 2.1. The whole idea for solving the system of equations $F(\mathbf{x}) = 0$ where $F : R^n \to R^n$ is to start from an initial point $\mathbf{x}_0$ and approximate the function $F(\mathbf{x})$ by a linear function at this point. Then compute a new point $\mathbf{x}_1 = \mathbf{x}_0 + \Delta\mathbf{x}$. Repeating this process. Under certain conditions it can be shown that the sequence of points generated by this schema converges to a root of $F(\mathbf{x})$.

In order to move from $\mathbf{x}_0$ to $\mathbf{x}_0 + \Delta\mathbf{x}$, we can use the relation $[\mathrm{DF}(\mathbf{x}_0)]\Delta\mathbf{x} = -F(\mathbf{x}_0)$, where $\mathrm{DF}(\mathbf{x}_0)$ denotes the matrix of partial derivatives of $F(\mathbf{x})$ defined by $\left[\left(\frac{\partial F_i}{\partial \mathbf{x}_j}\right)_{i,j}\right]$ at point $\mathbf{x}_0$.For the case of linear programming, the matrix $\mathrm{DF}(\mathbf{x}_0)$ can be stated as follow

$$\mathrm{DF} = \begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ D_x & 0 & D_z \end{pmatrix}.$$

The last equation in 2.1 may be expressed in either of the following forms:

$$\begin{array}{l} a.\mathbf{z}_i - \frac{\mu}{\mathbf{x}_i} = 0 \\ b.\mathbf{x}_i - \frac{\mu}{\mathbf{z}_i} = 0 \\ c.\mathbf{x}_i\mathbf{z}_i - \mu = 0 \end{array} \tag{2.3}$$

which gives the following forms for D_x and D_z respectively:

1. $D_x = \mu\mathrm{Diag}(1/\mathbf{x}_1^2, \ldots, \mathbf{x}_n^2)$, $D_z = I$, and $\mathrm{RHS} = -(\mathbf{z}_0 - \frac{\mu}{\mathbf{x}_0})$.
2. $D_x = I$, $D_z = \mu\mathrm{Diag}(1/\mathbf{z}_1^2, \ldots, \mathbf{z}_n^2)$, and $\mathrm{RHS} = -(\mathbf{x}_0 - \frac{\mu}{\mathbf{z}_0})$.
3. $D_x = \mathrm{Diag}(\mathbf{z}_1, \ldots, \mathbf{z}_n)$, and $D_z = \mathrm{Diag}(\mathbf{x}_1, \ldots, \mathbf{x}_n)$, $\mathrm{RHS} = -(\mu - \mathbf{x}_0 \odot \mathbf{z}_0)$ [1] .

On each iteration of Newton's method we have to solve a system of the form

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ D_x & 0 & D_z \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \\ \Delta\mathbf{z} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mathrm{RHS} \end{pmatrix}.$$

Now consider the SDP problem

$$\begin{array}{ll} \min & C \bullet X \\ \text{s.t.} & \mathcal{A}\mathbf{vec}X = b \\ & X \succeq 0. \end{array}$$

[1] $a \odot b = (a_1 b_1, \ldots, a_n b_n)$

Applying the appropriate barrier function yields to the new problem

$$\begin{array}{lll} \mathbf{P}_\mu : & \min & C \bullet X - \mu \ln \mathrm{Det}(X) \\ & \text{s.t.} & \mathcal{A}\mathbf{vec}X = \mathbf{b} \\ & & X \succeq 0 \end{array}$$

With the Lagrangian

$$L(X, \mathbf{y}) = C \bullet X - \mu \ln \mathrm{Det}(X) + \mathbf{y}^T(b - \mathcal{A}\mathbf{vec}X)$$

Using the fact that $\frac{\partial \ln \mathrm{Det}(X)}{\partial X_{ij}} = \frac{1}{\mathrm{Det}(X)} \frac{\partial \mathrm{Det}(X)}{\partial X_{ij}} = \frac{(-1)^{i+j}}{\mathrm{Det}(X)} \mathrm{det}_{ij}$, which is equal to (i, j)-th entry of X^{-1}, we get the following optimality conditions

$$\begin{array}{rl} \nabla_{\mathbf{y}} L(X, \mathbf{y}) = & \mathbf{b} - \mathcal{A}\mathbf{vec}X = 0 \\ \nabla_{\mathbf{x}} L(X, \mathbf{y}) = & C - \mu X^{-1} - \mathrm{Mat}\mathcal{A}^T\mathbf{y} = 0 \end{array}$$

which gives rise to the following system of equations

$$\begin{cases} \mathrm{Mat}\mathcal{A}^T\mathbf{y} + Z = C \\ \mathcal{A}\mathbf{vec}X = \mathbf{b} \\ Z - \mu X^{-1} = 0 \end{cases} \tag{2.4}$$

Analogous to the linear programming case, the primal central path can be defined as the set of positive semidefinite matrices that solve $\mathbf{P}_\mu$ for all values of $\mu \geq 0$.
Imposing the barrier function into dual the problem

$$\begin{array}{ll} \max & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} & \mathrm{Mat}\mathcal{A}^T\mathbf{y} + Z = C \\ & Z \succeq 0 \end{array}$$

gives rise to

$$\begin{array}{lll} \mathbf{D}_\mu : & \max & \mathbf{b}^t\mathbf{y} + \mu \ln \mathrm{Det}(Z) \\ & \text{s.t.} & \mathrm{Mat}\mathcal{A}^T\mathbf{y} + Z = C \\ & & X \succeq 0 \end{array}$$

with Lagrangian

$$L(\mathbf{y}, Z, X) = \mathbf{b}^T\mathbf{y} + \mu \ln \mathrm{Det}(Z) + X \bullet (C - \mathrm{Mat}\mathcal{A}^T\mathbf{y} - Z)$$

and optimality conditions

$$\begin{array}{l} \nabla_{\mathbf{y}} L(\mathbf{y}, Z, X) = \mathbf{b} - \mathcal{A}\mathbf{vec}X = 0 \\ \nabla_{\mathbf{z}} L(X, \mathbf{y}) = \mu Z^{-1} - X = 0 \\ \nabla_{\mathbf{x}} L(X, \mathbf{y}) = C - \mathrm{Mat}\mathcal{A}^T\mathbf{y} - Z = 0 \end{array}$$

or equivalently

$$\begin{array}{rl} 1. & \mathrm{Mat}\mathcal{A}^T\mathbf{y} + Z = C \\ 2. & \mathcal{A}\mathbf{vec}X = \mathbf{b} \\ 3.a. & Z - \mu X^{-1} = 0 \\ b. & X - \mu Z^{-1} = 0 \\ c. & XZ = \mu I \end{array} \tag{2.5}$$

There is a slight difference between 3.c of 2.5 and c in 2.3. We cannot use 3.c in this form to solve the system. The reason is that matrix multiplication is not commutative in general. For the case of our problem, unless current solution is not on the central path we cannot guarantee that $XZ = ZX$, and in the process of solving SDP we always try to stay close to central path and not exactly on it.
The main system that should be solved on each iteration of Newton's method can be represented as follows

$$\begin{pmatrix} \mathcal{A} & 0 & 0 \\ 0 & \mathcal{A}^T & I \\ D_X & 0 & D_Z \end{pmatrix} \begin{pmatrix} \mathbf{vec}\Delta X \\ \Delta\mathbf{y} \\ \mathbf{vec}\Delta Z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \text{RHS} \end{pmatrix}$$

Using 3.c. of 2.5 to derive D_X and D_Z for this system, and solving the system, shows that ΔZ is always symmetric, but it may not be the case for ΔX. Next time we will see that to solve this problem we can use different techniques like replacing ΔX by $\frac{\Delta X + (\Delta X)^T}{2}$ or replacing 3.c. in 2.5 with $X^{1/2} Z X^{1/2} = \mu I$ for $X = X^{1/2} X^{1/2}$, and then deriving D_X and D_Z. Intuitively any formulation of 3.c that keeps the left-hand side symmetric will imply a symmetric value for ΔX. Writing linear programming as a SDP problem does not create this problem because that for linear programming the matrices X and Z are going to be diagonal matrices and multiplication among diagonal matrices is commutative.

3 Suggested projects

3.1 KKT-Conditions for SDP. Derive a SD analogue of KKT-conditions, i.e. given a problem in following form

$$\begin{array}{ll} \min & f_0(\mathbf{x}) \\ \text{s.t.} & F(\mathbf{x}) \succeq 0 \quad (F : R^m \rightarrow R^{\frac{n\times n}{2}}) \end{array}$$

for $f_0(\mathbf{x})$ not necessarily a linear function, what are the optimality conditions for this problem.

3.2 Computing λ_2. Show that computing the second largest eigen value of a matrix is NP-hard.

3.3 Degeneracy in SDP. Study the degeneracy in the SDP. We know that if primal problem is nondegenrate then there exist a unique dual solution and if dual is nondegenrate then there is a unique primal solution, also we have already seen that uniqueness of both primal and dual problems implies strict complementarity, that is

$$\begin{cases} \text{Rank}(X) + \text{Rank}(Y) = n \\ \text{Rank}(X) \geq n - \sqrt[\underline{2}]{n^{\underline{2}} - m} \\ \text{Rank}(Y) \geq n - \sqrt[\underline{2}]{m} \end{cases} {}^{2}.$$

What are the sufficient conditions for uniqueness?

[2] $a^{\underline{2}} = \frac{a(a+1)}{2}$, and if $b = a^{\underline{2}}$ then $a = \sqrt[\underline{2}]{b}$